DTIC FILE COPY

AD-A220 459

COMPARISON OF SELECTION PROCEDURES AND VALIDATION OF CRITERION USED IN SELECTION OF SIGNIFICANT CONTROL VARIATES OF A SIMULATION MODEL

THESIS

James A. Gigliotti
Captain, USAF

AFIT/GOR/ENS/90M-7

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

90 04 13 218

(1)

AFIT/GOR/ENS/90M-7

COMPARISON OF SELECTION PROCEDURES AND VALIDATION OF CRITERION USED IN SELECTION OF SIGNIFICANT CONTROL VARIATES OF A SIMULATION MODEL

THESIS

James A. Gigliotti
Captain, USAF

AFIT/GOR/ENS/90M-7

AFIT/GOR/ENS/90M-7

COMPARISON OF SELECTION PROCEDURES AND VALIDATION OF CRITERION USED IN SELECTION OF SIGNIFICANT CONTROL VARIATES OF A SIMULATION MODEL

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Operations Research

James A. Gigliotti

Captain, USAF

March 1990

PREFACE

The purpose of this thesis was three-fold. The first purpose was to revise and extend the capabilities of existing software for selecting the significant control variables of a simulation model, based on a newly developed selection criterion. The second purpose was to compare the results obtained using the revised software employing two different selection procedures. And the third purpose was then to validate the effectiveness of the new selection criterion by comparison to results derived using other common selection criteria.

After extensive revision, the software, now renamed the Variable Subset Selection Program (VSSP), was ready for use. The VSSP was then used to evaluate data with known characteristics and data derived from an untested simulation model. The results obtained from this effort served to demonstrate the usefulness of the VSSP and the validity of the the new selection criterion. It is highly recommended that the work be continued, as further benefits are yet to be realized and may be of substantial significance.

The execution and preparation of this thesis would not have been possible without the help of others. I am deeply indebted to my faculty advisor, Major Kenneth Bauer, Jr., for his extensive time, patience, and assistance. I also wish to thank my thesis reader, Lt Colonel Thomas Schuppe, for pointing out my numerous writing errors and ensuring the final product was understandable. Finally, I wish to thank my family and friends for their continuous support and encouragement when the going got rough.

James A. Gigliotti

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The purpose of this thesis was three-fold. The first purpose was to revise and extend the capabilities of existing software for selecting the significant control variables of a simulation model, based on a newly developed selection criterion. The second purpose was to compare the results obtained using the revised software employing two different selection procedures. And the third purpose was then to validate the effectiveness of the new selection criterion by comparison to results derived using other common selection criteria.

Extensive revision of the existing software was necessary to prepare it for use. Initially, the software was revised to extend its adaptability to evaluating new data and to increase user friendliness. Next, a new procedure was added to the software to permit it to evaluate data using a Stepwise (Forward Selection) procedure. Previously, the software only performed analysis of the data through an Enumerated Subsets approach. After revision of the software was complete, it was renamed the Variable Subset Selection Program (VSSP).

Once the VSSP was ready, it was used to evaluate two types of data. The first type of data was created using a known stochastic structure. Three sets of this data was used, each set using a different covariance structure between the responses and control variables. The second type of data was created from an untested simulation model. This data provided a means of validating the program and the selection criterion incorporated into it. In addition, the data derived from the untested simulation model was also evaluated using a commercially available statistical software package employing several common selection

criterions. These results were then compared to those obtained using the VSSP.

Overall, this study found that as the amount of data evaluated by the VSSP increased, any differences between the control variables selected as significant, by either the Enumerated Subsets or Stepwise procedure, disappeared. In fact, a point was apparently reached where additional data caused no change in the results obtained. Also, when the covariances between the control variables are known, this only makes any difference when a minimal amount of data is available. And finally, comparison of the results obtained by the VSSP and the commercial software package showed the new criterion to be comparable to those commonly in use. The new criterion also had the advantage of not requiring a subjectively determined stopping criteria for selecting the significant control variables, unlike some of the other criterion in use today.

The recommendations made from this study involved further work on the VSSP and additional experimentation which can be performed to extend the usefulness of the new criterion. Several suggestions for enhancements to the VSSP, primarily in regards to adding additional evaluation procedures and increasing program efficiency, were noted. There is also much work remaining in regards to the new selection criterion. One possibility mentioned, would be to see if the point where further data provides no additional benefit to the evaluation, could be analytically determined.

COMPARISON OF SELECTION PROCEDURES AND VALIDATION OF CRITERION USED IN SELECTION OF SIGNIFICANT CONTROL VARIATES OF A SIMULATION MODEL

Background

When dealing with computer simulations it is typically desirable to have a general understanding of how the simulation inputs will affect the final results. It is also desirable to be able to accurately estimate the expected simulation response. Furthermore, if the estimation of the response can be achieved with a subset of the simulation inputs (variables), a variance reduction on the estimator of the mean can also be realized. One way of achieving these goals is through the identification of a good subset of control variates. Control variates, also known as control variables, are variables which have a significant covariance with the response of interest.

The development of a quick and easy method for identifying the subset of significant control variates in a simulation model would greatly decrease the time and effort required to gain insights into the simulation. Identifying the significant control variates for a simulation model can also enhance the process of preparing and implementing an experimental design. It would eliminate the guesswork in determining which variables to concentrate on in a subsequent experimental design. This could also save computer time by identifying a subset of the available control variates to work with, since the standard experimental design requires 2^k simulation runs to acquire data, where k is the number of variables being tested.

The need for research into this problem and several methods for approaching it have been identified in current literature, but little substantial work has yet been accomplished (Bauer, 1987:2). Furthermore, Pritsker (1986:748) notes that even though theoretical development of control variates has proceeded, little practical application has been reported.

Specific Problem

The purpose of this thesis was to compare selection procedures for selecting the significant control variates of a simulation model and to validate the selection criterion used.

The scope of this thesis was confined to revising and adding new procedures to previously written software for identifying the significant control variates of a simulation model and applying the software to selected data sets. The data was also evaluated using commercial software and additional selection criteria. The results were then used to compare the selection procedures employed and to validate the selection criterion.

Sub-objectives

In order to solve the specific problem the following sub-objectives or steps were accomplished. The first sub-objective was to revise and incorporate new procedures into existing software for evaluating simulation output and identifying the significant control variates.

The second sub-objective was to test the revised software on several sets of simulation model output with known responses, significant control variates, and covariances between the control

variates. This output is referred to as the control data/output in later text.

The third sub-objective was to compare the sets of significant control variates identified by the revised software with each other and with those known for the control data/output.

The fourth sub-objective was to use the revised software to identify the significant control variates of an untested simulation model. This simulation model is referred to as the test model or data generation model in later text.

And the fifth sub-objective was to compare the selected control variates with variable subsets selected using commercially available software and various other selection criteria.

General Methodology

For each of the sub-objectives outlined above, there were associated sets of methods and equipment required to accomplish them. Accomplishment of the first sub-objective, to revise and incorporate new procedures into the existing software for evaluating simulation output and identifying the significant control variates, necessitated a two-fold approach. The first approach involved revising existing software previously developed by Bauer (1989B). The basis for the software was an evaluation of all possible combinations (enumerated subsets) of the control variates involved. To successfully perform this task required a thorough understanding of the underlying logic and statistical concepts on which the software was based. The next step was the actual revision of the existing software with the goals of increasing generality and user-friendliness of the software.

The second approach to meeting this sub-objective dealt with adding a new routine to the revised software based on a stepwise evaluation procedure and the same statistical accept/reject criterion as the enumerated subset routine. Again, study was necessary to understand the stepwise procedure and construct the logic for implementation. When that step was completed, the actual stepwise procedure software was written, debugged, and incorporated into the overall revised program. The resulting software product is referred to as the Variable Subset Selection Program (VSSP) in later text.

The second sub-objective, test the Variable Subset Selection Program on several sets of simulation model output with known responses, significant control variates, and covariances between the control variates, was completed as follows. The first step was to obtain data/output with these characteristics. Next, this data/output was evaluated using the Variable Subset Selection Program. The data/output was evaluated using both the enumerated subsets and stepwise procedures incorporated into the program. The end result of this sub-objective was a single subset of control variables derived using each of the selection procedures.

The third sub-objective, compare the sets of significant control variates identified by the Variable Subset Selection Program with each other and with those known for the control data/output, was straightforward and involved answering a series of questions. Were there any differences in the number of control variates identified as significant? Were there any differences in the specific control variates identified as significant?

The fourth sub-objective, use the Variable Subset Selection Program to identify the significant control variates of an untested simulation model, was completed as follows. The first step was to obtain a simulation model for evaluation. Next, the simulation model was run on the AFIT VMS computer system to create the output data to use as input data for the Variable Subset Selection Program. And finally, the Variable Subset Selection Program evaluated the model output and a subset of the significant control variates was selected. The difference between the data/output derived from this model and the data/output used in sub-objective two is that the significant control variates had not been previously determined.

The fifth and final sub-objective, compare the selected control variates with variable subsets selected using commercially available software and various other selection criteria, was accomplished as follows. First, the SAS statistical package, installed on the AFIT VMS system, was selected as representative of commercially available software. Next, the SAS procedures for Enumerated Subsets, and Stepwise (using Forward Selection, Backward Selection, and R^2 Maximization [MAXR] options) evaluation was applied to the data/output of the untested simulation model. And finally, the results obtained using SAS were compared to those obtained using the Variable Subset Selection Program. The primary purpose of this sub-objective was to validate the criterion used by the VSSP and demonstrate it will provide comparable results.

Thesis Organization and Development

A review of literature relevant to this thesis is presented in

Chapter II. The literature review covers the topic of control variates and their theoretical development. Also reviewed are the common selection criteria and selection procedures in use today, and the theoretical development of a new selection criterion.

In Chapter III the detailed methodology used in approaching and completing this thesis is covered. Then, the results of the research are presented and discussed in Chapter IV. And finally, the conclusions and recommendations reached, after evaluating the data, are given in Chapter V.

II. LITERATURE REVIEW

The following discussion is a review of the literature that has relevance to this thesis topic.

Variance Reduction Techniques

The need for some form of Variance Reduction Technique becomes apparent when it is understood that simulation is an experimental technique, for analyzing systems which usually involve the use of stochastic processes (Tomick,1988:1.7). Since stochastic processes are "a collection of random variables" (Ross, 1985:72), then the output from a simulation experiment is also a random variable. Thus, the response of interest is only an estimate of the true value. From Pritsker (1986:742), "the variance of the sample mean is a derived measure of the reliability that can be predicted if a simulation experiment is repeatedly performed". Pritsker also states that "Variance Reduction Techniques (VRTs) are methods that attempt to reduce the estimated values of variance through the setting of special conditions or through the use of prior information."

In a survey of Variance Reduction Techniques (VRTs), performed by Wilson (1984:280), VRTs are divided into two categories: correlation methods, and importance methods. His paper discusses three correlation methods (common random numbers, antithetic variates, and control variates) and four importance methods (importance sampling, conditional Monte Carlo, stratified sampling, and systematic sampling).

The basic difference between the two categories is the underlying

principle of the methods. The correlation methods increase the efficiency of the simulation by exploiting linear correlations among the simulation responses and input variables. The importance methods achieve variance reduction by concentrating on prior knowledge of the input domain.

Of the VRT methods discussed by Wilson, this thesis concentrates on the use of control variates. The rationale for this decision were two-fold. First, this is a promising technique which can provide valuable insights into the problem, if even to identify a lack of correlation between the inputs (i.e. control variates) and responses. And second, even though theoretical development has been proceeding, not much has been done in the way of practical applications. Further information on this method follows.

Control Variates

The method of control variates, also known as control variables, is one of the correlation methods mentioned previously. Basically, "... the control variates technique uses regression methods to exploit any inherent correlation between an output and a selected random variable vector with known mean that is observed on each run" (Wilson, 1984:280).

The remainder of the discussion on control variates will cover the types of control variates, the theory behind the concept of control variates, and a summary of recent work accomplished on this topic.

Types of Control Variates. Law and Kelton (1982:359) define two types of control variates. The two types are internal, or concomitant, control variates, and external control variates. The first type of control variate addressed is the internal control variate. Internal

control variates may be selected from among any of the input random variables, or simple functions of them, since their means are known. This view is further endorsed by Bauer (1989A:0-63), when he states "any input random variable is a candidate for a control variate." In addition, analysis of the simulation's use of the input random variables should identify at least the sign of the correlation with the output random variable. An advantage of internal control variates is they must typically be generated anyway during a simulation and therefore add essentially nothing to the cost of running the simulation.

The second type of control variates, external control variates, require the simultaneous simulation of a similar, but analytically tractable, system using common random numbers. The corresponding output from this similar simulation is then used as the control variate. By analytically tractable, it is meant that the expected value of the output variable can be calculated exactly. It is then hoped that the similar nature of the tractable simulation will induce a correlation between the two outputs, which can then be exploited. The major disadvantage of this type of control variate is that it requires a second simulation model and additional simulation runs, so it is not costless.

Theory of Control Variates. To restate, "the concept associated with control variates is the identification of a variable, say X, that has a positive covariance with the variable of interest, say Y" (Pritsker, 1986:748).

Unless noted otherwise, the following theory is based on a class handout provided by Bauer (1989A).

Univariate Simulation Response with a Single Control. Assume that Y is an unbiased estimator of the response of interest θ; that is, $E(Y) = \theta$, where E(Y) is the expected value of Y. Let X be an input random variable, selected as the control variate. It is further assumed that X has a known expected value of u_X and is highly correlated with Y. Then, for any constant b (known as the control coefficient), the controlled estimator Y(b), given by Eq (1), is unbiased for θ.

$$Y(b) = Y - b(X - u_X) \tag{1}$$

Then the variance of Y(b) is

$$Var[Y(b)] = Var(Y) + b^2 Var(X) - 2bCov(Y,X) \tag{2}$$

From review of Eq (2), it is readily apparent that a variance reduction can be achieved if

$$2bCov(Y,X) > b^2 Var(X) \tag{3}$$

So, if the condition of Eq (3) is met, then the controlled estimator will have a smaller variance then the uncontrolled estimator. It is also apparent that if the variables X and Y are independent, in which case $Cov(Y,X) = 0$, then no improvement over the uncontrolled estimator is possible. Next, with the application of some calculus to Eq (2), the optimal control coefficient, ß, for which the variance of Y(b) is a minimum, is given by

$$ß = Cov(Y,X)/Var(X) \tag{4}$$

Substituting Eq (4) into Eq (1) leads to Eq (5) which gives the optimal

controlled estimator $Y(\beta)$

$$Y(\beta) = Y - [Cov(Y,X)/Var(X)] * (X - u_X) \quad (5)$$

And substituting Eq (4) into Eq (2) yields the corresponding minimum variance for $Y(\beta)$ of

$$Var[Y(\beta)] = (1 - p_{XY}^2) * Var(Y) \quad (6)$$

where p_{XY} is the correlation coefficient between X and Y. Therefore as the absolute value of p_{XY} tends to its maximum value of one, the variance of $Y(\beta)$ decreases.

For the next step, let θ be denoted by u_Y. Then the average of the controlled observations Y_i, for i = 1 to K, is an unbiased point estimator of u_Y. This estimator is represented by Eq (7).

$$\bar{Y}(\beta) = (1/K) \sum_{i=1}^{K} Y_i(\beta) \quad (7)$$

where K is the sample size and

$$Y_i(\beta) = Y_i - \beta(X_i - u_X) \quad (8)$$

Typically, the optimal value β is unknown and must be estimated. However, β can be estimated as follows:

> An intuitive estimate of β replaces the right-hand side of Eq (4) with the appropriate sample quantities. This solution turns out to be the least squares solution for β. When the assumption of joint normality between Y and X is made, then the least squares solution is also the maximum likelihood solution. (Bauer, 1987:6)

So the following equation provides an estimate of β.

$$\hat{\beta} = \sum_{i=1}^{K} (Y_i - \bar{Y}) * (X_i - \bar{X}) / \sum_{i=1}^{K} (X_i - \bar{X})^2 \quad (9)$$

where

$$\bar{Y} = \sum_{i=1}^{K} Y_i / K \tag{10}$$

and

$$\bar{X} = \sum_{i=1}^{K} X_i / K \tag{11}$$

The point estimate of u_Y is

$$\bar{Y}(\hat{\beta}) = \bar{Y} - \bar{\beta}(\bar{X} - u_X) \tag{12}$$

Then, the variance of the point estimator is given by

$$\hat{Var}[\bar{Y}(\hat{\beta})] = \hat{Var}[Y(\hat{\beta})]/K \tag{13}$$

where

$$\hat{Var}[Y(\hat{\beta})] = (1-\hat{p}_{XY}{}^2) * \hat{Var}(Y) \tag{14}$$

Bauer (1987:6) provides the derivation of the interval estimate through the application of regression theory and assuming that Y and X are jointly normal random variables. The resulting 100(1-a)% confidence interval is given by

$$\bar{Y}(\hat{\beta}) \pm t_{K-2}(1-a/2) * (\hat{Var}[\bar{Y}(\hat{\beta})] * s_{11})^{1/2} \tag{15}$$

where

$$s_{11} = \sum_{i=1}^{K} (X_i - u_X)^2 / K \sum_{i=1}^{K} (X_i - \bar{X})^2 \tag{16}$$

t_{K-2} is the Student's t-distribution with K-2 degrees of freedom, and

'a' is the desired significance level.

Since ß is estimated, the variance reduction achieved is smaller then could have been obtained had the optimal control coefficient been known. This loss of variance reduction is quantified as the Loss Factor (LF). The loss factor is defined as 'the ratio of the variance of the estimator of u_y when the optimal control coefficient is not known to the variance of the estimator when the coefficient is known' (Bauer, 1987:9). Bauer (1987:10) provides the derivation of the loss factor, which reduces to

$$LF = (K-2)/(K-Q-2) \qquad (17)$$

where

Q = the number of controls
K = the number of independent replications

Furthermore, the 'loss factor acts as a multiplier to the minimum variance ratio (MVR)' (Bauer, 1987:10), which is given by

$$MVR = Var[\bar{Y}(ß)]/Var(\bar{Y}) \qquad (18)$$

The MVR represents the possible variance reduction when the optimal control coefficient is known. Multiplying Eq (17) and Eq (18) together leads to the variance ratio (VR). The VR represents the possible variance reduction when ß is estimated.

$$VR = LF * MVR \qquad (19)$$

Univariate Simulation Response with Multiple Controls. Kleijnen (1974:151) addresses the extension of theory to multiple control variates. Also, Bauer provides a summary of 'the development

presented by Lavenberg and Welch (1981) for simulation output analysis based on independent replications, batch means, and regenerative analysis" (Bauer, 1987: 11).

Let Y be the univariate response with variance v_y^2, $\tilde{X}$ be the Qx1 vector of controls, $\tilde{v}_{xy}$ be the Qx1 vector of covariances between Y and $\tilde{X}$, and $\tilde{E}_x$ be the QxQ covariance matrix of the controls. Then rewriting Eq (12) with multiple controls leads to

$$\bar{Y}(\hat{\tilde{ß}}) = \bar{Y} - \hat{\tilde{ß}}^T(\bar{\tilde{X}} - \tilde{u}_x) \tag{20}$$

where $\hat{\tilde{ß}}$, $\bar{\tilde{X}}$, and $\tilde{u}_x$ are Qx1 vectors. The vector of optimal control coefficients is given by

$$\tilde{ß} = \tilde{E}_x^{-1}\tilde{v}_{xy} \tag{21}$$

Since the covariance matrices are usually unknown, ß can be estimated by substituting the sample values of $\tilde{E}_x$ and $\tilde{v}_{xy}$ into Eq (21). This leads to the following equation:

$$\hat{\tilde{ß}} = \tilde{S}_x^{-1}\tilde{S}_{xy} \tag{22}$$

where $\tilde{S}_x^{-1}$ is the inverse of the QxQ sample covariance matrix of the controls. and $\tilde{S}_{xy}$ is the Qx1 vector of sample covariances between the univariate response and the vector of controls.

Assuming that Y and $\tilde{X}$ have a joint multivariate normal distribution, then

$$\begin{vmatrix} Y \\ \tilde{X} \end{vmatrix} \sim N_{Q+1} \left\| \begin{vmatrix} \tilde{u}_y \\ \tilde{u}_x \end{vmatrix} , \begin{vmatrix} \tilde{v}_y^2 & \tilde{v}_{yx} \\ \tilde{v}_{xy} & \tilde{E}_x \end{vmatrix} \right\| \tag{23}$$

Consequently, $\tilde{Y}(\tilde{ß})$ is unbiased for u_y and an exact 100(1-a)% confidence interval is given by

$$\bar{\tilde{Y}}(\tilde{\tilde{ß}}) \pm t_{K-Q-1}(1-a/2)D{*}S_{yx} \tag{24}$$

where

$$D^2 = K^{-1} + (K-1)^{-1}(\tilde{\bar{X}} - \tilde{u}_x)^T\tilde{S}_x^{-1}(\tilde{\bar{X}} - \tilde{u}_x) \tag{25}$$

$$S_{yx}^2 = (K-Q-1)^{-1}(K-1)(S_y^2 - \tilde{S}_{xy}^T\tilde{S}_x^{-1}\tilde{S}^{-1}\tilde{S}_{xy}) \tag{26}$$

t_{K-Q-1} is the Student's t-distribution with (K-Q-1) degrees of freedom, and S_y^2 is the sample variance of Y (Bauer et al, 1988:3).

<u>Multiple Simulation Responses with Multiple Controls</u>. Bauer, Venkatraman, and Wilson (1987:334) provide the necessary theoretical structure for handling the case of P response variables and Q control variates. When dealing with multiple variables, $\tilde{Y}$ is a Px1 vector of response variables, $\tilde{ß}$ is a PxQ matrix of control coefficients, and $\tilde{S}$ is the sample covariance matrix of the response vector. Assuming that $\tilde{Y}$ and $\tilde{X}$ have a joint multivariate normal distribution, then

$$\begin{vmatrix} \tilde{Y} \\ \tilde{X} \end{vmatrix} \sim N_{Q+1} \left\| \begin{vmatrix} \tilde{u}_y \\ \tilde{u}_x \end{vmatrix}, \begin{vmatrix} \tilde{E}_y & \tilde{E}_{yx} \\ \tilde{E}_{xy} & \tilde{E}_x \end{vmatrix} \right\| \tag{27}$$

Consequently, $\tilde{\bar{Y}}(\tilde{\tilde{ß}})$ is an unbiased estimator of $\tilde{u}_y$ and an exact 100(1-a/2)% confidence ellipsoid for $\tilde{u}_y$ is given by

$$[\tilde{\bar{Y}}(\tilde{\tilde{ß}}-\tilde{u}_y]^T\tilde{S}_{yx}^{-1}[\tilde{\bar{Y}}(\tilde{\tilde{ß}})-\tilde{u}_y] \leq P(K-Q-1)(K-P-Q)^{-1}DF(1-a;P,K-P-Q) \tag{28}$$

where

$$D^2 = K^{-1} + (K-1)^{-1}(\tilde{\bar{X}} - \tilde{u}_x)^T\tilde{S}_x^{-1}(\tilde{\bar{X}} - \tilde{u}_x) \tag{29}$$

$$\tilde{S}_{yx}^{2} = (K-Q-1)^{-1}(K-1)(\tilde{S}_y - \tilde{S}_{yx}\tilde{S}_x^{-1}\tilde{S}_{xy}) \qquad (30)$$

and $F(1-a;n_1,n_2)$ is the F-distribution with n_1 and n_2 degrees of freedom (Bauer et al, 1987:335).

"The advantage of the above approach over selecting separate controls for each response is the capability to form a joint confidence region for the response vector, rather than being limited to univariate confidence intervals" (Tomick, 1988:2.10).

Selection of Significant Control Variates

Neter states that "One of the most difficult problems in regression analysis often is the selection of the set of independent variables to be employed in the model" (1983:417). Regardless of the problem involved, there are several reasons to restrict the number of variables used in a model: (1) A model with a large number of variables can be expensive to maintain, (2) Models with a limited number of variables are easier to analyze and understand, and (3) The presence of many highly intercorrelated variables may add little to the predictive power of the model, detracting from the model's descriptive abilities and increasing the problem of roundoff error (Neter, et al; 1983:418).

The selection of the significant control variates depends primarily on the selection criteria and selection procedure used. The selection criteria determines the relative significance of a regression variable (control variate) and this may vary as the criteria varies. The type of selection procedure has an effect on whether the subset(s) of control variates chosen is the "best" subset or is a "near-best" subset.

Selection Criteria. The most common selection criteria in use

today and which are reviewed here, are R_p^2, R_a^2, and C_p. In addition, a new selection criteria, BC_p, is also presented. Any of these selection criteria can be used for selecting one or more variable subsets.

In the discussion of each criteria, the following notation is used. P is the number of potential parameters. The intercept term, β_0, counts as one parameter, so there are P-1 potential X variables $(X_1, \ldots, X_{P-1})$. p is the number of parameters present in a subset, so any subset regression model contains p-1 X variables. And n is the number of observations.

<u>The R_p^2 Criteria</u>. R_p^2 is based on the coefficient of multiple determination R^2, for a subset of size p, and is defined as:

$$R_p^2 = 1 - (SSE_p / SSTO) \qquad (31)$$

where

SSE_p = Error sum of squares for a parameter subset of size p.
SSTO = Total sum of squares for y.

SSTO is equivalent to SSE_1 which is the regression model with only an intercept term. SSTO remains constant for each subset evaluated, so as p increases, R_p^2 increases. This occurs since SSE_p can not increase as additional variables are added to the model. Consequently, R_p^2 reaches a maximum when all P-1 variables enter the regression model. Therefore the intent is not to maximize R_p^2, but to find a point where adding additional variables to the model does not increase R_p^2 significantly. "Clearly, the determination of where diminishing returns sets in is a judgmental one" (Neter, et al; 1983:422).

<u>The R_a^2 Criteria</u>. The adjusted coefficient of multiple determination, R_a^2, is very similar to R_p^2, and is defined as:

$$R_a^2 = 1 - ((n-1/n-p)*(SSE_p/SSTO)) \quad (32)$$

However, unlike R_p^2, this criterion 'takes the number of parameters in the model into account through the degrees of freedom' (Neter, et al; 1983:424). Therefore, while seeking the maximum value of R_a^2, it is possible for this value to decrease as p increases if the reduction in SSE_p is too small to offset the loss of a degree of freedom.

The C_p Criteria. The C_p criteria is based on minimizing the total mean squared error of the n fitted values for each of the various subset regression models. C_p is defined as:

$$C_p = (SSE_p \,/\, MSE(X_1,\ldots,X_{P-1})) - (n - 2*p) \quad (33)$$

where

$MSE(X_1,\ldots,X_{P-1})$ = mean squared error of the model with all P parameters.

The above equation assumes that the model which includes all P parameters provides an unbiased estimate of the variance. In the event the model used has substantial bias, it may be best to expand the set of potential variables to eliminate the bias.

In using the C_p criterion, identification of an appropriate subset of X variables is based on: (1) A small value of C_p, and (2) A C_p value near p. When plotting C_p vs. p, models with little bias will fall near the line C_p = p, and models with significant bias will be substantially above the line.

The BC_p Criteria. The BC_p, or Best Controls, criteria is a new criteria developed by Bauer and Wilson (1990). Unless otherwise

noted, the following discussion is based on their paper.

Development of the selection criteria is presented for two cases. The first case is where the covariance matrix, of variables and responses, is estimated. The second case is where the covariance matrix is known.

Before proceeding, the 'best' subset of variables is defined as that subset which produces a confidence region of minimum expected volume. Consequently, the selection criteria is designed to identify which subset of variables will produce this result.

Nomenclature. Let $Y = (Y_1, \ldots, Y_p)'$ denote a column vector of p responses generated on a single run of a simulation model whose mean response $\tilde{u}_y = E(Y)$ is to be estimated. Furthermore, let $C = (C_1, \ldots, C_q)'$ denote a column vector of q concomitant control variates with known mean $\tilde{u}_c = E(C)$ and let b denote a fixed (p x q) matrix of control coefficients, then the controlled response

$$Y(b) = Y - b(C - \tilde{u}_c) \tag{34}$$

is an unbiased estimator of $\tilde{u}_y$ whose dispersion can be minimized by the appropriate choice of b. Let $\tilde{E}_y$, $\tilde{E}_c$, and $\tilde{E}_{yc}$ respectively denote the covariance matrices of Y, C, and between Y and C; then

$$\tilde{E}_y = \mathrm{Cov}(Y) = E[(Y - \tilde{u}_y)(Y - \tilde{u}_y)'], \tag{35}$$

$$\tilde{E}_c = \mathrm{Cov}(C) = E[(C - \tilde{u}_c)(C - \tilde{u}_c)'], \tag{36}$$

and

$$\tilde{E}_{yc} = \mathrm{Cov}(Y,C) = E[(Y - \tilde{u}_y)(C - \tilde{u}_c)']. \tag{37}$$

And then, in terms of these quantities, the conditional covariance matrix of Y given C = c is

$$\tilde{E}_{y|c} = \mathrm{Cov}(Y|C = c) = \tilde{E}_y - \tilde{E}_{yc} - \tilde{E}_{yc}\tilde{E}_c^{-1}\tilde{E}_{yc}' \tag{38}$$

for every c which is an element of R^q (i.e. $c \in R^q$).

Rubinstein and Marcus (1985) showed that the generalized variance of Y(b) is minimized by the optimal matrix of control coefficients

$$\tilde{ß} = \tilde{E}_{yc}\tilde{E}_c^{-1} \tag{39}$$

Typically, $\tilde{E}_{yc}$ is unknown so $\tilde{ß}$ must be estimated. Let k denote the number of independent replications of the simulation to be performed; and for $j = 1,\ldots,k$, let (Y_j, C_j) denote the results observed on the j^{th} run. Then, in terms of the statistics

$$\bar{Y} = (1/k) \sum_{j=1}^{k} Y_j, \tag{40}$$

$$S_y = (1/(k-1)) \sum_{j=1}^{k} (Y_j - \bar{Y})(Y_j - \bar{Y})', \tag{41}$$

$$\bar{C} = (1/k) \sum_{j=1}^{k} C_j, \tag{42}$$

$$S_c = (1/(k-1)) \sum_{j=1}^{k} (C_j - \bar{C})(C_j - \bar{C})', \tag{43}$$

and

$$S_{yc} = (1/(k-1)) \sum_{j=1}^{k} (Y_j - \bar{Y})(C_j - \bar{C})'; \tag{44}$$

the sample analogue of $\tilde{ß}$ is

$$\hat{\tilde{ß}} = S_{yc}S_c^{-1} \tag{45}$$

Thus the j^{th} controlled response is estimated as $Y_j(\hat{\tilde{ß}}) = Y_j - \hat{\tilde{ß}}(C_j - \tilde{u}_c)$ for $j = 1,\ldots,k$; and the overall controlled point estimator of $\tilde{u}_y$ is

$$\bar{Y}(\hat{\tilde{\beta}}) = (1/k) \sum_{j=1}^{k} Y_j(\hat{\tilde{\beta}}) = \bar{Y} - \hat{\tilde{\beta}}(\bar{C} - \tilde{u}_c) \tag{46}$$

Bauer and Wilson (1990:4) also state that "In large scale simulation experiments, frequently the responses and the controls are jointly normal because these statistics are simultaneously accumulated over the duration of each run and thus are subject to a central-limit effect;... Thus it is reasonable to assume that Y and C jointly possess a multivariate normal distribution." Thus,

$$\begin{vmatrix} Y \\ C \end{vmatrix} \sim N_{p+q} \left\| \begin{vmatrix} \tilde{u}_y \\ \tilde{u}_c \end{vmatrix}, \begin{vmatrix} \tilde{E}_y & \tilde{E}_{yc} \\ \tilde{E}_{cy} & \tilde{E}_c \end{vmatrix} \right\| \tag{47}$$

with $\det(\tilde{E}_y) \neq 0$ and $\det(\tilde{E}_c) \neq 0$, where $\tilde{u}_c$ is known but $\tilde{E}_y$, $\tilde{E}_{yc}$, and possibly $\tilde{E}_c$ are unknown. Applying the basic results of Rao (1967) in this situation, Bauer and Wilson then compute a confidence ellipsoid for $\tilde{u}_y$ as follows. Let

$$T^2(\tilde{u}) = [\bar{Y}(\hat{\tilde{\beta}}) - \tilde{u}]'[d'd\hat{\tilde{E}}_{y|c}]^{-1}[\bar{Y}(\hat{\tilde{\beta}}) - \tilde{u}] \tag{48}$$

for every $\tilde{u}$ which is an element of R^p, where

$$M = \begin{vmatrix} (C_1 - \bar{C})' \\ \cdot \\ \cdot \\ \cdot \\ (C_k - \bar{C})' \end{vmatrix}, \quad d = (1/k)1_k - M(M'M)^{-1}(\bar{C} - \tilde{u}_c), \tag{49}$$

$$\hat{\tilde{E}}_{y|c} = (k-1 \;/\; k-q-1)(S_y - S_{yc}S_c^{-1}S_{yc}'), \tag{50}$$

and let 1_k denote a k-dimensional column vector of ones. Conditioned on the values of the controls $\{C_j : j = 1,\ldots,k\}$ observed across all k runs, $T^2(\tilde{u}_y)$ has Hotelling's T^2-distribution with k-q-1 degrees of freedom;

therfore an exact unconditional 100(1-a)% confidence ellipsoid for $\tilde{u}_y$ is

$$M(q;k,p,a) = \{\tilde{u} \in R^p : (T^2(\tilde{u}) \;/\; k-q-1) * (k-q-p \;/\; p) \leq F_{1-a}(p,k-q-p)\}, \tag{51}$$

where $F_{1-a}(p,k-q-p)$ is the quantile of order 1-a for an F-distribution with p and k-q-p degrees of freedom.

Selection Criteria for Estimated Covariance Matrix. Given the replication count k, the p-dimensional estimand $\tilde{u}_y$, and the confidence coefficient 'a', the goal is to select a subset of controls from a set of q control-variate candidates such that the resulting controlled confidence-region estimator for $\tilde{u}_y$ analogous to Eq (51) is, in some sense, as 'small' and as 'stable' as possible. Bauer and Wilson formulate such an estimator with some additional notation. Given a nonnegative integer r representing the number of control-variate candidates currently under consideration, let $u(q,r) = q!/[r!(q-r)!]$ be the number of distinct control-variate subsets of size r. Then, for r = 0,...,q and h = 1,...,u(q,r) let I(h,r) denote the h^{th} distinct subset of size r from the set {1,...,q}. Furthermore, on the j^{th} run of the simulation model, let $C_j(h,r)$ denote the r-dimensional vector of controls corresponding to the index-set I(h,r)

$$C_j(h,r) = [C(i_1 j),\ldots,C(i_r j)]', \tag{52}$$

where $i_1 < \ldots < i_r$ and $\{i_1,\ldots,i_r\} = I(h,r)$.

Similarly, let $\tilde{E}_{y|c}(h,r)$, $\hat{\hat{\beta}}(h,r)$, $\bar{Y}[\hat{\hat{\beta}}(h,r)]$, d(h,r), and $\hat{\hat{E}}_{y|c}(h,r)$ respectively denote the analogues of Eqs (38), (45), (46), (49), and (50) when the control vector C(h,r) defined by I(h,r) is used to compute

the controlled estimator of $\tilde{u}_y$. Corresponding to Eq (48), Bauer and Wilson then derive

$$T^2(\tilde{u},h,r) = \{\bar{Y}[\hat{\tilde{ß}}(h,r)] - \tilde{u}\}'[d'(h,r)d(h,r)\hat{\tilde{E}}_{y|c}(h,r)]^{-1} *$$
$$\{\bar{Y}[\hat{\tilde{ß}}(h,r)] - \tilde{u}\}; \quad (53)$$

and the exact 100(1-a)% confidence ellipsoid for $\tilde{u}_y$ analogous to Eq (51) as

$$M(h,r;k,p,a) = \{\tilde{u}\in R^p: (T^2(\tilde{u},h,r) \ / \ k-r-1) * (k-r-p \ / \ p)$$
$$\leq F_{1-a}(p,k-r-p)\}. \quad (54)$$

Then the size of the confidence region is calculated by

$$V(h,r;k,p,a) = (|\hat{\tilde{E}}_{y|c}(h,r)|^{1/2} \ / \ (p/2)G(p/2)) *$$
$$\{[d'(h,r)d(h,r)](pi*p*(k-r-1 \ / \ k-r-p) *$$
$$F_{1-a}(p,k-r-p)\}^{p/2} \quad (55)$$

where G, for this and following equations, denotes the Gamma function. Also the mean volume of the confidence region, Eq (54), is given by

$$E[V(h,r;k,p,a)] = w(h,r;k,p,a) \ ([G(k/2)]^{1/2} \ / \ G[(k-p)/2]), \quad (56)$$

where

$$w(h,r;k,p,a) = ([|\tilde{E}_{y|c}(h,r)|G(k/2)]^{1/2} \ / \ (p/2)G(p/2)) *$$
$$[2*pi*p*F_{1-a}(p,k-r-p) \ / \ k(k-r-p)]^{p/2}. \quad (57)$$

And the mean square volume of the confidence region, Eq (54), is

$$E[V^2(h,r;k,p,a)] = (w^2(h,r;k,p,a) \ / \ G[(k-2p)/2]) *$$
$$\prod_{i=1}^{p} [(k-r-i)/(k-r-2i)]. \quad (58)$$

Although alternative expressions are available for the mean volume and mean square volume, Bauer and Wilson believe their equations (56) and (58) are easier to use from both a mathematical and computational standpoint.

In the case of a univariate response, Nelson (1989) and Schmeiser (1982) define the standard measures of confidence-interval stability as the standard deviation (SD) and coefficient of variation (CV) of the confidence-interval half-length; and in the case of a multivariate response, the corresponding stability measures, based on the confidence-region volume of Eq (55), are

$$SD[V(h,r;k,p,a)] = w(h,r;k,p,a)\ \{(1 / G[(k-2p)/2]) * \prod_{i=1}^{p} [(k-r-i)/(k-r-2i)] - (G(k/2) / G^2[(k-p)/2])\}^{1/2} \qquad (59)$$

and

$$CV[V(h,r;k,p,a)] = \{(G^2[(k-p)/2] / G(k/2)G[(k-2p)/2]) * \prod_{i=1}^{p} [(k-r-i)/(k-r-2i)] - 1\}^{1/2}. \qquad (60)$$

Similar to Schmeiser's (1982) conclusions about the performance of univariate confidence intervals, Bauer and Wilson (1990) observed that a confidence-region estimator M(h,r;k,p,a) with large values of (59) or (60) will give false signals about the intrinsic precision of the associated point estimator $\bar{Y}[\hat{\beta}(h,r)]$ in a large percentage of applications. Bauer and Wilson then went on to state

> Thus it seems reasonable to select a control vector C(h,r) that yields a small value for (59) or (60). However, it would be undesirable to base a control-variate selection criterion exclusively on the principle of minimizing (59) or (60) -- this

principle fails to exclude confidence-region estimators that achieve smaller values of the standard deviation or coefficient of variation of the volume simply by increasing the mean volume. On the other hand, it is also undesirable to base a control-variate selection criterion exclusively on the principle of minimizing the mean volume without attempting simultaneously to reduce or at least bound the standard deviation or coefficient of variation of the volume. (1990:6)

Bauer and Wilson wanted to ensure that the delivered confidence-region estimator Eq (54) is both small and stable, so they proposed a control-variate selection criterion based on the principle of minimizing the mean square volume given by Eq (58). Since $E[V^2(h,r;k,p,a)] \geq E^2[V(h,r;k,p,a)] \geq 0$, it is clear that their selection criterion will tend to reduce the mean volume at least indirectly; and in comparison to a selection criterion based on minimization of the mean volume, Eq (56), the selection criterion will yield smaller values of the standard deviation Eq (59) and the coefficient of variation Eq (60) of the confidence-interval volume; unless both procedures select exactly the same control variates with probability one. In the event both procedures do select the same control variates, the two selection criteria yield identical results. Therefore their strategy of basing the criterion on minimizing the mean square volume of the delivered confidence-region estimator offers many of the advantages of selection criteria based on minimizing the mean, standard deviation, or coefficient of variation of the delivered volume without some of the potential disadvantages of these latter selection criteria.

To implement the proposed selection criterion in practice, it is necessary to minimize the mean square volume $E[V^2(h,r;k,p,a)]$ as a function of h and r, where $r = 0,\ldots,q$ and $h = 1,\ldots,u(q,r)$. Since $|\tilde{\Sigma}_{Y|C}(h,r)|$ is generally unknown, this quantity is replaced by the

unbiased estimator

$$|\hat{\hat{E}}_{y|c}(h,r)| \prod_{i=1}^{p} [(k-r-1)/(k-r-i)] \qquad (61)$$

to obtain the expression that must be minimized in selecting the final subset of control variates

$$\text{MIN } \{(|\hat{\hat{E}}_{y|c}(h,r)|G(k/2) \;/\; [(p/2)G(p/2)]^2 G[(k-2p)/2])$$
$$* \; [(2*pi*p*F_{1-a}(p,k-r-p) \;/\; k(k-r-p)]^p$$
$$* \; \prod_{i=1}^{p} [(k-r-1)/(k-r-i)]\} \qquad (62)$$

subject to the constraints of $0 \le r \le q$ and $1 \le h \le u(q,r)$. Let r^* and h^* denote the optimal values of r and h in Eq (62). The delivered point and confidence-region estimators of $\tilde{u}_y$ are given by $\bar{Y}[\hat{\hat{ß}}(h^*,r^*)]$ and $M(h^*,r^*;k,p,a)$, respectively. Thus Eq (62) gives the selection criteria for the case where the covariance matrix is unknown.

Selection Criteria for Known Covariance Matrix. Often, situations arise in discrete event simulation where the covariance matrix of some set of control variates is known analytically or can be readily evaluated by numerical methods. In this situation an alternative to the estimator $\hat{\hat{ß}}$, for the unknown covariance matrix case, for the optimal control coefficient vector $\tilde{ß}$ can be obtained by replacing S_c, in Eq (45), with $\tilde{E}_c$ to obtain

$$\overset{\text{\textasteriskcentered}}{ß} = S_{yc}\tilde{E}_c^{-1}. \qquad (63)$$

In this case the controlled point estimator of $\tilde{u}_y$ has the form

$$\bar{Y}(\overset{\text{\textasteriskcentered}}{ß}) = \bar{Y} - \overset{\text{\textasteriskcentered}}{ß}(\bar{C} - \tilde{u}_c). \qquad (64)$$

Under the assumption of joint normality, Bauer (1987) proved that $\bar{Y}(\overset{*}{\beta})$ is an unbiased estimator of $\tilde{u}_y$ with covariance matrix

$$\overset{*}{\tilde{E}} = \mathrm{Cov}[\bar{Y}(\overset{*}{\beta})] = (k-2 / k(k-1))\tilde{E}_{y|c} + (q+1 / k(k-1))\tilde{E}_y. \qquad (65)$$

To derive an approximate 100(1-a)% confidence region for $\tilde{u}_y$ centered at $\bar{Y}(\overset{*}{\beta})$, Bauer and Wilson began with an unbiased estimator of $\overset{*}{E}$ obtained by replacing the unknown covariances on the right-hand side of Eq (65) with the corresponding sample covariances

$$\dot{S} = (k-2 / k(k-1))\hat{\tilde{E}}_{y|c} + (q+1 / k(k-1))\hat{\tilde{E}}_y. \qquad (66)$$

Provided that $k >> q$, Eq (66) implies that $\dot{S}$ is approximately independent of $\bar{Y}(\overset{*}{\beta})$ and possesses the p-dimensional central Wishart distribution with k-q-1 degrees of freedom on the covariance matrix $\tilde{E}_{y|c}$. Then

$$\dot{T}^2(\tilde{u}) = [\bar{Y}(\overset{*}{\beta}) - \tilde{u}]'\dot{S}^{-1}[\bar{Y}(\overset{*}{\beta}) - \tilde{u}] \qquad (67)$$

for every $\tilde{u}$ which is an element of R^p. Thus for large k, $\dot{T}^2(\tilde{u}_y)$ has an approximate Hotelling's T^2-distribution with k-q-1 degrees of freedom; and in this case an approximate 100(1-a)% confidence ellipsoid for $\tilde{u}_y$ is given by

$$\dot{M}(q;k,p,a) = \{\tilde{u} \in R^p : (\dot{T}^2(\tilde{u}) / k-q-1)(k-q-p / p) \le F_{1-a}(p,k-q-p)\}. \qquad (68)$$

Progress to this point parallels the development of equations (54) through (62) that apply to the situation where the covariance matrix is unknown. Bauer and Wilson then sought a control vector C(h,r) which

minimized the mean sqaure volume of the confidence ellipsoid. For $r = 0, \ldots, q$ and $h = 1, \ldots, u(q,r)$, let $\tilde{\beta}(h,r)$, $\bar{Y}[\tilde{\beta}(h,r)]$, $\tilde{E}(h,r)$, and $\dot{S}(h,r)$ respectively denote the analogues of Eqs (63), (64), (65), and (66) when the control vector $C(h,r)$ defined by the index-set $I(h,r)$ is used to compute the controlled estimator of $\tilde{u}_y$. Equation (67), then becomes

$$\dot{T}^2(\tilde{u},h,r) = \{\bar{Y}[\beta^*(h,r)] - \tilde{u}\}'[\dot{S}(h,r)]^{-1}\{\bar{Y}[\beta^*(h,r)] - \tilde{u}\} \tag{69}$$

for every $\tilde{u}$ which is an element of R^p; and the approximate $100(1-a)\%$ confidence ellipsoid for $\tilde{u}_y$ analogous to Eq (68) is

$$\dot{M}(h,r;k,p,a) = \{\tilde{u} \in R^p : (\dot{T}^2(\tilde{u},h,r) \;/\; k-r-1)(k-r-p \;/\; p) \le F_{1-a}(p,k-r-p)\}. \tag{70}$$

Now the confidence region of Eq (70) has volume

$$\dot{V}(h,r;k,p,a) = (|\dot{S}(h,r)|^{1/2} \;/\; (p/2)G(p/2)) * [pi*p*(k-r-1) \;/\; k-r-p)F_{1-a}(p,k-r-p)]^{p/2} \tag{71}$$

Then, assuming $\dot{S}(h,r)$ approximately possesses the p-dimensional Wishart distribution with k-r-1 degrees of freedom on the covariance matrix $E^*(h,r)$, Bauer and Wilson derived the following approximate expression for the mean square volume

$$E[\dot{V}^2(h,r;k,p,a) \simeq (|E^*(h,r)| \;/\; [(p/2)G(p/2)]^2) * [pi*p \;/\; k-r-p)F_{1-a}(p,k-r-p)]^p \prod_{i=1}^{p} (k-r-i). \tag{72}$$

To implement their proposed selection criterion for this new linear control-variates estimation procedure, it is necessary to minimize Eq

(72) as a function of h and r in a manner similar to that for Eq (62). Since $|\dot{S}(h,r)|$ is generally unknown, this quantity is replaced by the unbiased estimator

$$|\dot{S}(h,r)| \prod_{i=1}^{p} [(k-r-1)/(k-r-i)] \tag{73}$$

as in Eq (61) to obtain the expression that must be minimized in selecting the final subset of control variates

$$\text{MIN } \{(|\dot{S}(h,r)| \;/\; [(p/2)G(p/2)]^2) \;* \\ [pi*p*(k-r-1)*F_{1-a}(p,k-r-p) \;/\; k-r-p]^p\} \tag{74}$$

subject to the constraints of $0 \leq r \leq q$ and $1 \leq h \leq u(q,r)$. If r^{**} and h^{**} denote the optimal values of r and h in Eq (74) then the delivered point and confidence-region estimators of $\tilde{u}_y$ are $\bar{Y}[\hat{\beta}(h^{**},r^{**})]$ and $\dot{M}(h^{**},r^{**};k,p,a)$, respectively. Thus Eq (74) gives the selection criteria for the case where the covariance matrix is unknown.

Selection Procedures. There are a variety of procedures available for selection of the best or near-best subset(s) of variables. The procedures presented here are: enumerated subsets, stepwise regression, forward selection and backward elimination, and regression by leaps and bounds.

Each procedure has its own advantages and disadvantages. However, the primary rationale for using selection procedures other than an enumerated subsets, or all-possible subsets, approach is to reduce the amount of computations required. Even for a moderate number of variables the number of subsets to evaluate becomes $2^P - 1$, where P is the number of variables being considered for the model. In addition,

only the enumerated subsets approach will assure that the 'best' subset(s) of variables will be selected, based on the selection criteria used.

Enumerated Subsets. The enumerated subsets, or all-possible subsets, approach is based on the following algorithm.

1) A regression model with no X variables (i.e. $Y_i = \beta_0 + e_i$) is evaluated using the selected criteria.

2) A series of regression models including each variable, individually, are evaluated using the selected criteria.

3) A series of regression models including each possible pair of variables are evaluated. This process continues, increasing the number of variables in the model one at a time, until a model incorporating all variables is reached. Again, each model is evaluated using the selected criteria.

4) Based on the criteria employed (i.e. R_p^2, R_a^2, C_p, BC_p, etc.), the best, or k best, subset(s) are selected.

Stepwise Regression. The stepwise regression procedure is the most common search method used when the number of variables involved make an enumerated subsets approach infeasible.

> Essentially, this search method develops a sequence of regression models, at each step adding or deleting an X variable. The criterion for adding or deleting an X variable can be stated equivalently in terms of error sum of squares reduction, coefficient of partial correlation, or F^* statistic. (Neter, et al; 1983:430)

For the search algorithm which follows, the F^* statistic is used for illustration of the concepts involved.

1) A regression model is fitted for each of the X variables. For each regression model the F^* statistic is obtained as follows:

$$F_k^* = MSR(X_k) \;/\; MSE(X_k) \tag{75}$$

The X variable with the largest F^* value is selected to enter the model. Provided the F^* value exceeds a predetermined level required to enter the model. If all of the F^* values are below the threshold level, the search ends.

2) If an X variable, say X_i, enters the model, then all regression models with two variables are fitted; where X_i is one of each pair. For each regression model, the partial F test statistic (Neter, et al; 1983:289) is obtained.

$$F_k^* = MSR(X_k \mid X_i) \;/\; MSE(X_i, X_k) = (b_k \;/\; s(b_k))^2 \tag{76}$$

Where b_k and $s(b_k)$ are the estimated regression coefficient of variable k and the estimated variance of the estimated regression coefficient of variable k, respectively. Again, the X variable with the largest F^* value enters the model, if it exceeds the threshold level. Otherwise the procedure ends.

3) When more then one variable enters the model, it is then determined if any of the variables in the model should be dropped. The F^* values are derived as follows:

$$F_k^* = MSR(X_k \mid X_i, \ldots, X_j) \;/\; MSE(X_k, X_i, \ldots, X_j) \tag{77}$$

Where X_k is the variable being tested for possible elimination from the model and $X_i, \ldots, X_j$ are the other variables in the model. The X variable with the smallest F^* value is selected to exit the model. Provided the F^* value falls below a predetermined level required to

exit the model. If all of the F^* values are above the threshold level, all variables remain in the model.

4) Steps (2) and (3) are repeated until no further X variables can meet the threshold levels to enter or exit the model. It should be noted that as the number of variables in the model increases, the size of the subsets evaluated in step (2) increases. Where each subset evaluated includes the variables currently in the model, plus one of the variables not in the model.

Forward Selection and Backward Elimination. Both forward selection and backward elimination procedures are simplified variations of the stepwise regression procedure. The forward selection procedure differs from the stepwise regression procedure by not testing a variable, once it has entered the model, for possible elimination from the model.

The backward elimination procedure is the opposite of the forward selection procedure. This procedure begins with the model containing all the X variables. Then the F^* value for each variable is calculated and the variable with the smallest value identified. If this value is less than the threshold level, it is dropped from the model. This process continues until no further variables can be dropped from the model.

From this overview of control variates theory and methods for selecting significant variables, a methodology was developed for proceeding with the work involved. This methodology is presented in the following chapter.

III. METHODOLOGY

The overall objective of this thesis was to develop software to assist in identifying the significant control variables in a simulation model. The software was to incorporate a newly developed selection criteria in conjunction with several common selection procedures. After this was accomplished, the objective moved to applying the software to a simulation model and evaluating the results.

Selection and Preparation of Computer Code

The first step in proceeding was to select and prepare the necessary computer code/software. The software specifically is the Variable Subset Selection Program, the simulation model which provided data to test it on, and software for processing the data collected from the simulation model.

Variable Subset Selection Program. The Variable Subset Selection Program was developed from a previously written program. Before the software was ready for use, extensive revision and expansion of the code was performed. The primary goals achieved in revising the software dealt with increasing the ease of use of the software, and permitting either manual or external file data entry. Prior to these revisions, the program data had to be coded directly into the software before it could be compiled and executed.

Expansion of the software dealt with the addition of a capability to perform a stepwise regression procedure in conjunction with the new selection criteria. This was in addition to the enumerated subsets

procedure originally incorporated.

Data Generation Model. Additionally, a simulation model was selected to generate data for testing the variable subset selection software. It was decided to select a model which simulates the operation of a Local Area Network (LAN) and the interaction of the system peripherals. The network model, on which this simulation is based, is commonly found in simulation journals and serves as a practical benchmark for testing purposes.

The LAN simulation model consists of seven nodes and is illustrated in Figure 3.1. Node 1 represents the terminals connected to the LAN system. Commands to the LAN system are generated according to an exponential distribution. The commands are received at node 2, which serves as a delay buffer and simulates the possibility of all system peripherals being busy and unable to accept new commands. Next the commands move to node 3, which routes the command to a system peripheral. The routing of a command from a node to any other node in the system is based on probabilities contained in a probability matrix. A command routed from node 3 can go to node 1, 4, 5, 6, or 7. The queue capacity of nodes 4, 5, 6, and 7 is infinite and the queue discipline is First-In-First-Out (FIFO). At nodes 4, 5, 6, and 7 the service times are distributed exponentially. Once a command is through being serviced by a peripheral unit, it returns to node 3, where it is routed to another peripheral for further processing or sent to node 1. When a command returns to node 1, the time it took to get through the system is noted and the command terminated. The SLAM and associated FORTRAN code for this simulation model, along with the following post-processing

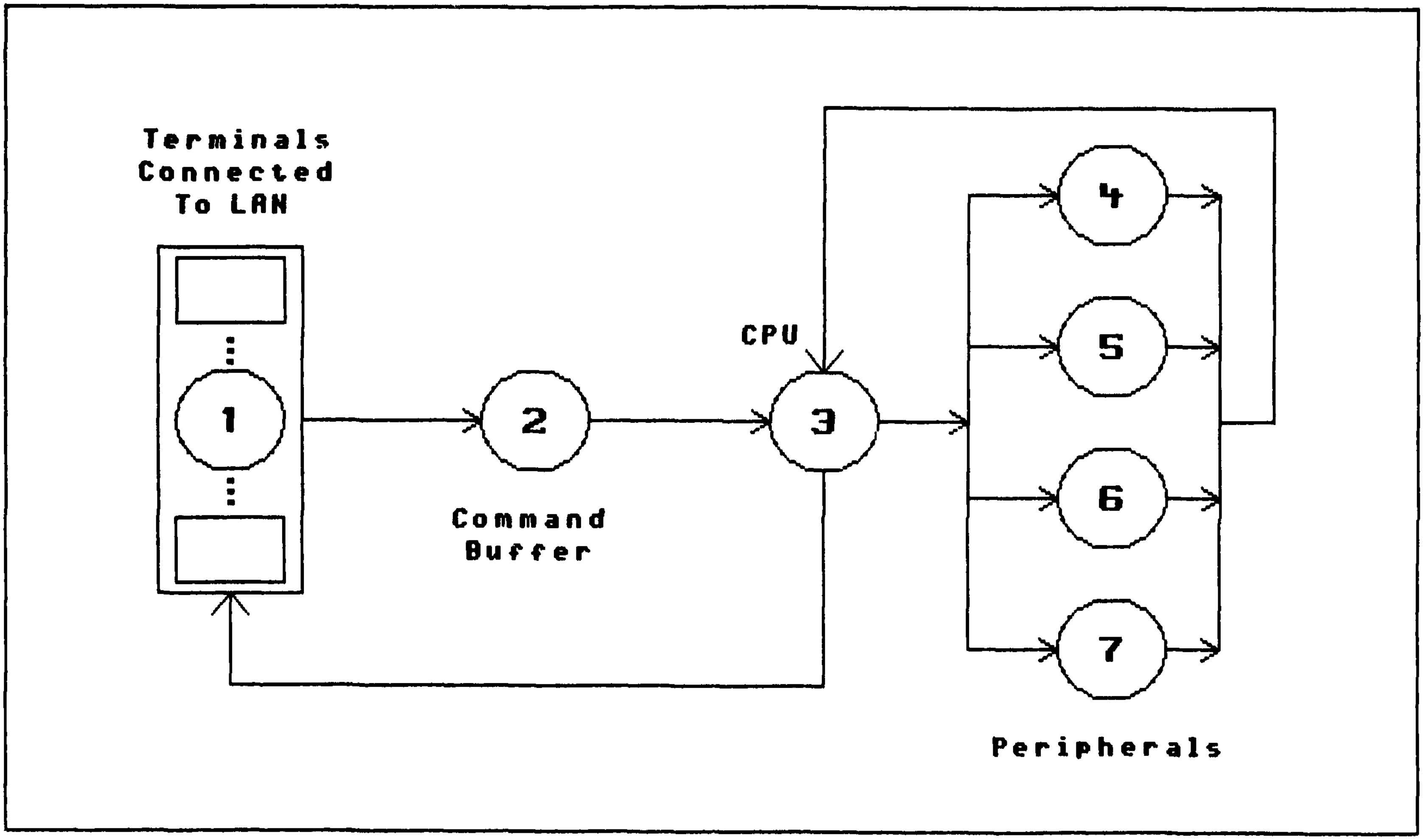


Figure 3.1 LAN Simulation Model Network

code, are provided in Appendix C.

The data collected using this model was related to the number of commands through the system, average utilization and total service time at each system node, number of departures from a node, and number of departures from node 3 to the other system nodes. This data was then processed for use by the Variable Subset Selection Program.

Data Post-Processing Software. After collection of output data from the Data Generation Model, data post-processing was performed. The purpose of this post-processing was to convert the control variate data into the form of work and routing variables. These types of variables are desirable because of the characteristics they possess. The desired characteristics are a sample mean of zero and sample variance of one.

The post-processing software performed this conversion according to the following equations

$$X_k^*(t) = (f(k,t)^{-1/2} / w_k(f(t))) \sum_{j=1}^{f(k,t)} (U_j(k) - u_k) / s_k \quad (78)$$

where

$f(k,t)$ = number of service times that are finished at node k during time period $(0,t)$
w_k = relative frequency with which a customer visits node k
$U_k(j)$ = the j-th service time at node k
$u_k = E[U_k(j)]$
$s_k = Var[U_k(j)]$

and

$$X_k = \sum_{j=1}^{N(t)} [(U_k(j) - p_k(*)) / (N(t)(1 - p_k(*))p_k(*))^{-1/2}] \quad (79)$$

for $j = 1,\ldots,S$ and $N(t) > 0$

where

$p_k(*)$ = probability of transition from node 3 to node k
$U_k(j)$ = flag whether or not transition j was to node k. If so, then $U_k(j)=1$, otherwise it equals zero.
$N(t)$ = total number of transitions from node 3 up to time t
S = total number of nodes in the LAN model.

Equation (78) applied to the work variables and Eq (79) to the routing variables. Additional information on work variables can be found in Lavenberg, et al (1982), and more information concerning routing variables in Bauer (1987).

The Response Variables

Next, a response vector was chosen for the analysis. The response chosen to form the response vector was the time it takes a command to pass through the LAN system. The time through the system begins when the command is issued at a terminal and ends when a command returns to node 1, informing it that the command has been executed.

The Control Variables

Next, the control variables were chosen, to provide a pool of variables for the Variable Subset Selection Program to select from. The control variables chosen were the total service times accumulated at nodes 1, 3, 4, 5, 6, and 7; and the number of departures from the CPU (node 3) to nodes 1, 4, 5, 6, and 7 respectively. This provided a pool of eleven variables.

The variables analogous to total service time at node 2, and number of departures from the CPU to node 2 and node 3 were not used. These variables were infeasible since their corresponding values were always zero. This resulted from the probability matrix associated with moving

from one node to another.

The selection of these variables also provided a mix of work and routing variables for the Variable Subset Selection Program to select from. The service time variables became work variables, and the departure variables became routing variables.

The Experimental Procedure

There were two distinct phases to the experimental procedure employed in this thesis effort. The first phase involved testing the Variable Subset Selection Program using data generated with known covariances between the control variables and known significant control variables. The second phase dealt with testing the Variable Subset Selection Program on output from an untested simulation model and evaluating this data/output using the VSSP and SAS software on the AFIT VAX system.

Evaluation Using Known Data. The known data was previously derived by Bauer and Wilson (1990) in testing of the original version of the selection software. A series of Monte Carlo experiments were performed to derive the data. Bauer and Wilson choose to use five control variate candidates and two responses. Two of the five control variates were constructed to be uncorrelated to the responses; therefore these control variates acted as decoys. Bauer and Wilson extended Eq (47) by partitioning the vector of control variates as $C = [X'\ W']$, where $X' = [C_1, C_2, C_3]$ consists of the three effective control variates and $W' = [C_4, C_5]$ consists of the decoy control variates. This resulted in the overall stochastic structure of

$$\begin{vmatrix} Y \\ X \\ W \end{vmatrix} \sim N_{2+3+2} \left\| \begin{vmatrix} u_y \\ u_x \\ u_w \end{vmatrix} , \begin{vmatrix} E_y & E_{yx} & 0 \\ E_{xy} & E_x & 0 \\ 0 & 0 & E_w \end{vmatrix} \right\| , \tag{80}$$

where 0 is an appropriately dimensioned matrix of zeros. Three versions of the covariance structure in Eq (78) were constructed. In all three cases

$$E_y = \begin{vmatrix} 1.0 & 0.3 \\ 0.3 & 1.0 \end{vmatrix}, \quad E_x = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}, \quad \text{and} \quad E_w = \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} \tag{81}$$

The cross-correlation matrix between the responses and effective control variates was of the form

$$E_{yx} = \begin{vmatrix} a & 0 & c \\ 0 & b & c \end{vmatrix} \tag{82}$$

The choices of a, b, and c for each case are summarized in Table 3.1.

Table 3.1 Cross-correlations in Eq (82)

	Case		
Parameter	I	II	III
a	0.8	0.5	0.2
b	0.5	0.5	0.2
c	0.5	0.5	0.2

Bauer and Wilson (1990)

One thousand sets of data were then derived from the normal distribution, Eq(80) for each case. Bauer and Wilson then evaluated the resultant data sets at k = number of replications = 10 and m = number of meta-experiments = 100, and k = 20 and m = 50 to obtain coverage and

volume reduction percentages for the delivered confidence regions.

Once the Variable Subset Selection Program was ready and the data sets obtained, evaluation was performed. Each data set was evaluated for the same k and m values, used by Bauer and Wilson, using both the enumerated subsets and stepwise procedures. In addition, for this thesis, further evaluation of the original data was performed at k = 50 and m = 20, and k = 1000 and m = 1. Again, evaluation was performed using both enumerated subsets and stepwise procedures.

Evaluation Using Untested Simulation Model. This section outlines the steps followed in generating the data, using the untested simulation model, for this thesis project. In generating the data/output from the untested simulation model, 1000 runs with different seeds for the random number generators were performed. The data generated from each run of the simulation was placed into an output file called 'DGM.OP'. The simulation model was run on the AFIT VAX system.

After the data/output was collected from the simulation model, the data was converted into the appropriate work and routing variables using the data post-processing software. As the work and routing variable data was created, it was placed in an output file called 'VSSP1.IN'. Also, a datafile called 'VSSP0.IN' was prepared which contained any other information required by the Variable Subset Selection Program. 'VSSP0.IN' and 'VSSP1.IN' then became data input files for the Variable Subset Selection Program.

Next, the Variable Subset Selection Program was run using data contained in the data input files 'VSSP0.IN' and 'VSSP1.IN'. The output of the software was then placed in a file called 'VSSP.OUT' and

contained the selected subset(s) of the control variables, deemed to be significant. After this point, the data evaluation phase occurred.

Data Evaluation Procedure

After the data had been generated and collected it then had to be evaluated. For the results obtained using the known data, the evaluation primarily consisted of comparing the results originally derived by Bauer and Wilson were compared to those derived using the VSSP. In addition, for results obtained from data runs beyond those performed by Bauer and Wilson, the results were examined to see what would happen to the results and if any noticeable trends would develop.

In regards to results obtained using the untested simulation model. The primary evaluation was based on comparison of the subsets selected as a result of each selection procedure employed, the BC_p criterion value for each resulting variable subset, and the estimated coverage and variance reduction associated with the control variables selected.

In addition, further comparison was performed against results derived using the SAS enumerated subsets and stepwise regression procedures. This primarily served the purpose of validating the results derived with the BC_p criterion.

The results derived from all these procedures are summarized and discussed in the following chapter.

IV. RESULTS AND DISCUSSION

The results obtained from this effort are presented and discussed according to phase of the experimental procedure in which they were derived. The phases dealt with evaluation of the data/output with known characteristics, and evaluation of data/output from the untested simulation model.

Results From Known Data

The results of the Variable Subset Selection Program runs, using the data/output with known characteristics, are presented in Tables 4.1, 4.2, 4.3, and 4.4 for the various combinations of k and m tested.

Examination of the results, using the data originally derived by Bauer and Wilson (1990), reveals several items of interest. First, as the number of replications increased, this had significant effects on the results. The percentage coverage and percentage confidence volume reduction figures stabilized as the number of replications increased. Also, for each data case, these figures become uniform, regardless of the evaluation method used and whether the covariances between the control variables was estimated or known. Apparently, some asymptotic point was reached where increasing the number of replications per meta-experiment ceased to make a difference.

It was also noted that for the initial results, for $k=10$ and $m=100$, the evaluations using the known covariance matrix had better coverage. However, this advantage quickly disappeared as the number of replications increased.

Table 4.1 VSSP results for k = 10, m = 100, and a = 0.10

Evaluation Method	Selection Criterion	% Coverage For Case I	II	III	% Vol Reduction For Case I	II	III
Enumerated Subsets	Eq (62)	78	76	69	96	83	74
	Eq (74)	94	88	81	83	75	72
Stepwise	Eq (62)	77	79	71	96	83	74
	Eq (74)	93	88	81	83	75	71

Table 4.2 VSSP results for k = 20, m = 50, and a = 0.10

Evaluation Method	Selection Criterion	% Coverage For Case I	II	III	% Vol Reduction For Case I	II	III
Enumerated Subsets	Eq (62)	84	80	82	95	76	65
	Eq (74)	82	84	84	88	74	64
Stepwise	Eq (62)	84	80	82	95	76	65
	Eq (74)	82	84	84	88	74	64

Next it was noticed that for the first two sets of results (k=10 & m=100, and k=20 & m=50), the results between the known and estimated covariance runs, appears to be converging. When the estimated covariances were used, coverage started out low and vice versa when the known covariances were used. However, later sets of results rebuked this observation. It is not yet understood why this occurred.

Finally, it was found that the volume reduction achieved was greater for data cases with greater covariances between the control variables and responses. This becomes more pronounced as the number of replications increase, but is still readily apparent even for a low

number of replications. Conversely, the covariance structure of the data/output had little, if any, impact on the coverage figures. This result makes intuitive sense since as covariances increase, the data/output becomes more tightly grouped; thus more data would fall within the calculated confidence volume.

Table 4.3 VSSP results for k = 50, m = 20, and a = 0.10

Evaluation Method	Selection Criterion	% Coverage For Case I	II	III	% Vol Reduction For Case I	II	III
Enumerated Subsets	Eq (62)	75	75	75	95	76	61
	Eq (74)	75	70	75	92	74	60
Stepwise	Eq (62)	75	75	75	95	76	61
	Eq (74)	75	70	75	92	74	60

Table 4.4 VSSP results for k = 1000, m = 1, and a = 0.10

Evaluation Method	Selection Criterion	% Coverage For Case I	II	III	% Vol Reduction For Case I	II	III
Enumerated Subsets	Eq (62)	100	100	100	95	76	59
	Eq (74)	100	100	100	95	76	59
Stepwise	Eq (62)	100	100	100	95	76	59
	Eq (74)	100	100	100	95	76	59

Additionally, the variable subsets selected by each evaluation method were compared for the various combinations of k and m tested. The purpose of this was to identify if any trends developed. The results are presented in Table 4.5.

Table 4.5 Percentage Number of Different Variable Subsets Selected; Comparison Between Enumerated Subsets and Stepwise Procedures

Evaluation Method	k=10, m=100			k=20, m=50		
	Case			Case		
	I	II	III	I	II	III
Enum. Subsets	5	7	4	0	2	0
Stepwise	3	3	4	0	2	2

Evaluation Method	k=50, m=20			k=1000, m=1		
	Case			Case		
	I	II	III	I	II	III
Enum. Subsets	0	0	0	0	0	0
Stepwise	0	0	0	0	0	0

From review of the results, it was obvious that as the number of replications per meta-experiment increased, both evaluation methods selected the same subsets. This tends to validate the results found earlier by reviewing the coverage and volume reduction figures. It is reasonable to find that as the coverage and volume reduction figures converge, the greater the similarity between the variable subsets selected.

Results From Untested Simulation Model Data

The results derived from evaluating the data/output from the untested simulation model are presented in Table 4.6. From comparing the variable subsets selected by VSSP and SAS, several items were noted. The most significant observation was that regardless of selection criterion used, almost all variable subsets were identical. It should

Table 4.6 Effects of various selection software and criterion on selection of variable subsets

Software Package	Evaluation Method	Selection Criterion	Criterion Value	Variable Subset* Selected
VSSP**	Enumerated Subsets	Eq (62)	3.156511	W1 W5 W6 R1 R4 R5 R6 R7
	Stepwise	Eq (62)	3.156511	W1 W5 W6 R1 R4 R5 R6 R7
SAS***	Enumerated Subsets	R_p^2	0.306733	W1 W5 W6 R1 R4 R5 R6 R7
		R_a^2	0.301137	W1 W5 W6 R1 R4 R5 R6 R7
		C_p	9.011377	W3 W4 W5 W6 R1 R4 R5 R6 R7
		S_p	1.196549	W1 W5 W6 R1 R4 R5 R6 R7
	Stepwise	R_p^2	0.306733	W1 W5 W6 R1 R4 R5 R6 R7
		C_p	7.062709	W1 W5 W6 R1 R4 R5 R6 R7
	Stepwise (Forward Selection)	R_p^2	0.306733	W1 W5 W6 R1 R4 R5 R6 R7
		C_p	7.062709	W1 W5 W6 R1 R4 R5 R6 R7
	Stepwise (Backward Selection)	R_p^2	0.306769	W1 W3 W4 W5 R1 R4 R5 R6 R7
		C_p	9.011541	W1 W3 W4 W5 R1 R4 R5 R6 R7
	Stepwise (MAXR)	R_p^2	0.306733	W1 W5 W6 R1 R4 R5 R6 R7
		C_p	9.011541	W1 W4 W5 W6 R1 R4 R5 R6 R7

* = W(i) is Work variable i, and R(i) is Routing variable i
** = Using a 90% level of significance (i.e. alpha = 0.10)
*** = Using the standard SAS default significance criterions

also be noted that of all the other selection criterion used in this comparison, S_p is the most closely related to BC_p and selected the same variable subset. Also, the R_a^2 criterion selected the same variable subset.

Other selection criterion used also provided comparable results. The R_p^2 criterion selected the same or nearly same variable subset depending on the point at which further improvement was discarded. If improvement required to be significant was set to ≥ 0.001 then the same variable subset was selected. However, if it was set to ≥ 0.01 then a subset without variable R1, was selected. As noted in the literature review, Chapter II, the level of significant improvement is highly subjective.

For the C_p criterion, only with the enumerated subsets procedure was the best variable subset found. In other procedures, where this selection criterion was available, a near-best variable subset was selected. Regardless of these differences, each variable subset selected contained one more variable than that selected using BC_p, and differed only slightly.

There are two final comments on the SAS derived results. First, not all selection criterion were available for all of the SAS procedures employed. And second, the variable subset selected by any procedure, may depend on the level of significance used. The default values for the SAS procedures, used in this evaluation, were not all set to the same level as used in the VSSP. This may account for the minor differences noted.

The conclusions drawn from these results are summarized in the

following chapter. This chapter also outlines any recommendations for further study and research in this area.

V. CONCLUSIONS AND RECOMMENDATIONS

From the work entailed in this thesis, the following conclusions and recommendations were derived.

Conclusions

From review and evaluation of the results, the following conclusions were drawn in regards to the Variable Subset Selection Program and the performance of the BC_p against other common criterion. The selection procedures and criterion incorporated into the Variable Subset Selection Program performed as well as expected. The results obtained by either selection procedure, enumerated subsets or stepwise (forward selection), are comparable and contain minimal variations, even when applied to a small number of replications. In addition, any differences between the results derived by either selection procedure decreases, as the number of replications increase, until there is no difference. It was also concluded that the known covariances between the controls only has a significant effect on the coverage and volume reduction figures when applied to a small number of replications. However, this advantage rapidly disappears as the number of replications increase.

In regards to the performance of the BC_p criterion in comparison to other criterion in common use today; the BC_p criterion provides reasonable and comparable results. Thus the Variable Subset Selection Program appears to be a reasonable substitute for use by organizations requiring limited evaluation of this sort, where use of a commercial package may be infeasible. This infeasibility may take several forms,

notably expense or cost-effectiveness, limited access to existing software, or time involved in learning to use the software effectively.

Recommendations

The recommendations derived from this study fall into two categories; 1) further improvements to the VSSP, and 2) further study/experimentation involving the BC_p criterion. There are numerous improvements which can be made to the VSSP. First and foremost is to implement a more efficient stepwise (forward selection) procedure. Primarily this entails developing a satisfactory scheme for saving prior pivots of the 'A' matrix so subsequent pivots are performed on the correct matrix. The current implementation performs a true stepwise (forward selection) search, but reverts to performing all pivots on each subset evaluated. This is inefficient and slows down program execution. The main benefit to be derived from this recommendation would be increased speed of evaluation.

Next, would be the implementation of additional selection procedures, notably a true stepwise procedure would be desirable. Other selection procedures as outlined in the Literature Review (Chapter 2) are also viable candidates.

Another recommendation is to revise the VSSP into a more modular and efficient form. This can be accomplished by breaking up the main program into subroutines. Each selection procedure implemented could be made a separate subroutine called by the main program when needed. Also, eliminate duplicate variables where possible without affecting program execution. And finally, arrays and matrices shared by other program subroutines could be incorporated into common blocks. This will

eliminate creation of duplicate constructs and cut down on the amount of memory required to run the program.

Additionally, there are several recommendations regarding further study, analysis, and research using the VSSP and the BC_p criterion. One recommendation is to evaluate output of a simulation model, using the VSSP, to obtain the single best variable subset. This could be done using both the enumerated and stepwise procedures to check for consistency in the results. Next, run two experimental designs, one using the variables selected by the VSSP and the second using all variables in the original full set. Then evaluate the experimental design results using similar selection procedures but based on other selection criterion (i.e. C_p, R^2, R_a^2, etc.). Compare the final results and evaluate the differences, if any. Does the variable subset selected by the VSSP provide a good starting point for an experimental design?

Another recommendation is to evaluate a set of data while increasing the number of replications, but holding the number of meta-experiments steady. How does this affect the overall coverage and volume reduction figures? Can any consistent trends be identified? The VSSP does not require the use of all data in a datafile to be used in execution.

And finally, experiment with data/output with known characteristics (i.e. means of controls and responses, and covariances between controls and responses), to find the point where the results converge across all combinations of selection procedures and whether the covariances are estimated or known. Then determine if this point can be arrived at analytically. If so, this type of knowledge could prevent gathering

more data than necessary for an optimal evaluation. This could also help contain costs associated with gathering data. Additionally, this may provide a means of assessing reliability of results, depending on the amount of data available for the evaluation.

APPENDIX A: FORTRAN Listing of Variable Subset Selection Program

```
C     PROGRAM SELECT(INPUT,OUTPUT,TAPE7,TAPE5=INPUT,TAPE6=OUTPUT)
C ********************************************************************
C *                                                                  *
C *                                                                  *
C *   This program provides both an 'all possible regressions'       *
C *   (i.e. enumerated subsets) and a stepwise (forward selection)   *
C *   approach to selecting the best subset of controls from a       *
C *   given candidate set. It assumes that a certain number of       *
C *   meta-experiments have been performed, each with the same       *
C *   number of replications.  Once the optimal subset has been      *
C *   identified, a confidence region is constructed about the mean  *
C *   vector for the responses.  The corresponding coverage and      *
C *   volume reduction are then tallyed and subsequently summarized. *
C *                                                                  *
C *   This program can be run in two modes. The user can either      *
C *   estimate the covariance matrix of controls or incorporate      *
C *   it directly. The program variable 'iknow' dictates which       *
C *   option is in effect (see code below).                          *
C *                                                                  *
C *   The program can also be run in the 'best m' regressions mode.  *
C *   ( Currently only configured for estimated covariance matrix    *
C *     of controls)                                                 *
C *   In other words it will compute the best m subsets of each      *
C *   possible subset size. This can be of interest if a single set  *
C *   of data is used.                                               *
C *                                                                  *
C *                                                                  *
C *   PROGRAM PARAMETERS:                                            *
C *                                                                  *
C *       Z1      = Max # of candidate controls allowed              *
C *       Z2      = Max # of responses allowed                       *
C *       Z3      = Z1 + Z2                                          *
C *       Z4      = Max # of best regressions which may be kept      *
C *                 (m in 'm best' as above)                         *
C *       Z5      = 2**Z1                                            *
C *       Z6      = Max # replications per meta experiment allowed   *
C *       Z7      = Max # of meta experiments allowed                *
C *       Z8      = (Z3*(Z3+1))/2                                    *
C *       MAX     = Maximum number of storage locations in array A   *
C *                 for matrices created by Subroutine GAUSS.        *
C *                                                                  *
C *   CORRESPONDING PRIMARY VARIABLES, INITIALIZED BY USER:          *
C *                                                                  *
C *       NX      = # of candidate controls                          *
C *       NY      = # of responses                                   *
C *       NVAR    = NX + NY                                          *
C *       KEEPERS = # of best regressions to be kept                 *
C *                 (m in 'm best' as above)                         *
```

```
C *       KNX     = 2**NX                                                *
C *       NUMREPS = # replications per meta experiment                   *
C *       META    = # of meta experiments                                *
C *                                                                      *
C *  NOTE:                                                               *
C *                                                                      *
C *       IN SUBROUTINE COVER : Z1 AND Z2 IN THE PARAMETER               *
C *                             STATEMENT, MUST BE SET TO                *
C *                             Z1 AND Z2, RESPECTIVELY, OF              *
C *                             THE MAIN PROGRAM PARAMETER               *
C *                             STATEMENT                                *
C *                                                                      *
C ************************************************************************
      PROGRAM SELECT
C
C ***************************************************************
C * PROGRAM PARAMETERS AND VARIABLE INITIALIZATION              *
C ***************************************************************
C
      INTEGER Z1, Z2, Z3, Z4, Z5, Z6, Z7, Z8, MAX
C
      PARAMETER (Z1=8, Z2=8, Z3=Z1+Z2, Z4=6, Z5=2**Z1)
      PARAMETER (Z6=50, Z7=100, Z8=((Z3*(Z3+1))/2))
      PARAMETER (MAX=50)
C
      COMMON /BLK1/ SIG, KK, IQQ, IP
C
      CHARACTER TITLE*25, RESPONS(Z2)*25, CONTROL(Z1)*25
      CHARACTER INFILE*25, OUTFILE*25, COVFILE*25
      CHARACTER ANSWER, XFILE*25
C
      INTEGER NK(Z3), MODELS(Z5,Z1), IBUFF(Z1), IX, KK, IP
      INTEGER IH(Z3), ICOVER(4), ICTOT(4), NBR(6), IIN, IQQ
      INTEGER NX, NY, NVAR, KEEPERS, KNX, NUMREPS, META
      INTEGER IKNOW, IWRITE, METHOD, I1, I2, IZ, KP, K
      INTEGER TMV, IND1(Z1+1), IND2(Z1+1), TIND(Z1+1)
C
      REAL COVCV(Z1,Z1), VECMUC(Z1), CBAR(Z1), VECCBAR(Z1)
      REAL FF(0:Z1), WKAREA(Z2), RSS(Z2,Z2), DUM(Z2)
      REAL TARGET(Z2,Z2), VECYBAR(Z2), VECMUY(Z2), YBAR(Z2)
      REAL A(MAX,Z3,Z3), VCV(Z8), FULL(Z3,Z3)
      REAL REGR(Z4,Z1,2), BUFF(Z4), BUFF2(Z4)
      REAL VR(2),VOLRED(2),COVERAG(4), X(Z6,Z3), TEMP(Z3)
      REAL XM(Z3), SUMDEV(2), SUMVU(2), SIG
C
C ***************************************************************
C *  BRIEF PROGRAM INTRODUCTION AND INITIAL DATA INPUT/OUTPUT   *
C *  ROUTINE                                                    *
C ***************************************************************
C
```

```
C ******************************************
C *  PROGRAM INTRODUCTION                   *
C ******************************************
C
      PRINT *, 'THIS IS THE VARIABLE SUBSET SELECTION PROGRAM'
      PRINT *, ''
      PRINT *, '   THIS PROGRAM IS DESIGNED TO EVALUATE OUTPUT'
      PRINT *, 'FROM A SIMULATION MODEL AND DETERMINE THE OPTIMAL'
      PRINT *, 'SUBSET OF VARIABLES, BASED ON THE ''BEST CONTROLS'''
      PRINT *, 'CRITERION.  YOU MAY CHOOSE EITHER AN ENUMERATED'
      PRINT *, 'SUBSETS OR STEPWISE (FORWARD SELECTION) APPROACH.'
      PRINT *, ''
      PRINT *, ''
C
C ******************************************
C * DATA INPUT                              *
C ******************************************
C
   10 PRINT *, 'DO YOU WISH TO ENTER PROGRAM DATA MANUALLY OR BY'
      PRINT *, 'DATAFILE?  ENTER M OR D:'
      READ *, ANSWER
      IF ((ANSWER.EQ.'D').OR.(ANSWER.EQ.'d')) THEN
         PRINT *, 'ENTER NAME OF THE INPUT DATAFILE: '
         READ *, INFILE
         OPEN(UNIT=10, FILE=INFILE, STATUS='OLD')
         READ(10,*) NX, NY, KEEPERS, NUMREPS, META
         READ(10,*) IKNOW, IWRITE, METHOD
         READ(10,*) SIG
         READ(10,*) (VECMUC(I),I=1,NX)
         READ(10,*) (VECCBAR(I),I=1,NX)
         READ(10,*) (VECMUY(I),I=1,NY)
         READ(10,*) (VECYBAR(I),I=1,NY)
         READ(10,*) TITLE
         READ(10,*) (CONTROL(I),I=1,NX)
         READ(10,*) (RESPONS(I),I=1,NY)
         IF (IKNOW.EQ.1) THEN
            READ(10,*) COVFILE
            IF (COVFILE.NE.INFILE) THEN
               OPEN(UNIT=15, FILE=COVFILE, STATUS='OLD')
               DO 15 I=1,NX
                  READ(15,*) (COVCV(I,J),J=1,NX)
   15          CONTINUE
               CLOSE(15)
            ELSE
               DO 20 I=1,NX
                  READ(10,*) (COVCV(I,J),J=1,NX)
   20          CONTINUE
            ENDIF
         ENDIF
         READ(10,*) XFILE
         OPEN(UNIT=20, FILE=XFILE, STATUS='OLD')
      ELSE
```

```
      IF ((ANSWER.NE.'M').AND.(ANSWER.NE.'m')) THEN
         PRINT *, 'INVALID INPUT, TRY AGAIN.'
         PRINT *, ''
         GOTO 10
      ENDIF
      PRINT *, 'ENTER THE FOLLOWING VARIABLE VALUES:'
      PRINT *, ''
      PRINT *, '  INPUT # OF CANDIDATE CONTROLS (PROGRAM '
      PRINT *, '  LIMIT = ',Z1,'):'
      READ *, NX
      PRINT *, '  INPUT # OF RESPONSES (PROGRAM LIMIT = ',Z2,'): '
      READ *, NY
      PRINT *, '  INPUT # OF BEST REGRESSIONS TO KEEP'
      PRINT *, '  (PROGRAM LIMIT = ',Z4,'): '
      READ *, KEEPERS
      PRINT *, '  INPUT # OF REPLICATIONS PER META EXPERIMENT'
      PRINT *, '  (PROGRAM LIMIT = ',Z6,'): '
      READ *, NUMREPS
      PRINT *, '  INPUT # OF META EXPERIMENTS DESIRED '
      PRINT *, '  (PROGRAM LIMIT = ',Z7,'): '
      READ *, META
      PRINT *, '  INPUT WHETHER COVARIANCE MATRIX OF CONTROLS IS'
      PRINT *, '  ESTIMATED (0), OR KNOWN (1): '
      READ *, IKNOW
      PRINT *, '  INPUT WHETHER YOU WANT THE META EXPERIMENT MODE '
      PRINT *, '  (0), OR THE BEST M REGRESSIONS MODE (1): '
      READ *, IWRITE
      PRINT *, 'INPUT WHETHER YOU WANT TO USE ENUMERATED SUBSETS'
      PRINT *, '(0), OR STEPWISE [FORWARD SELECTION] (1) METHOD: '
      READ *, METHOD
      PRINT *, '  INPUT LEVEL OF SIGNIFICANCE OF TEST (e.g. '
      PRINT *, '  90% = 0.90): '
      READ *, SIG
      PRINT *, '  INPUT THE KNOWN MEAN FOR EACH CONTROL: '
      PRINT *, ''
      DO 25 I=1,NX
         PRINT *, '       VECMUC(',I,') = '
         READ *, VECMUC(I)
25    CONTINUE
      PRINT *, '  INPUT AVERAGE OF INPUTS FOR EACH CONTROL: '
      PRINT *, ''
      DO 30 I=1,NX
         PRINT *, '       VECCBAR(',I,') = '
         READ *, VECCBAR(I)
30    CONTINUE
      PRINT *, '  INPUT ESTIMATED MEAN FOR EACH RESPONSE: '
      PRINT *, ''
      DO 35 I=1,NY
         PRINT *, '       VECMUY(',I,') = '
         READ *, VECMUY(I)
35    CONTINUE
```

```
      PRINT *, '  INPUT AVERAGE OF CONTROLLED OBSERVATIONS FOR '
      PRINT *, '  EACH RESPONSE: '
      PRINT *, ''
      DO 40 I=1,NY
         PRINT *, '      VECYBAR(',I,') = '
         READ *, VECYBAR(I)
40    CONTINUE
      PRINT *, '   INPUT ANALYSIS TITLE: '
      READ *, TITLE
      PRINT *, '  INPUT NAMES OF CANDIDATE CONTROLS: '
      PRINT *, ''
      DO 45 I=1,NX
         PRINT *, '      CONTROL(',I,') = '
         READ *, CONTROL(I)
45    CONTINUE
      PRINT *, '  INPUT NAMES OF RESPONSES: '
      PRINT *, ''
      DO 50 I=1,NY
         PRINT *, '      RESPONSE(',I,') = '
         READ *, RESPONS(I)
50    CONTINUE
      IF (IKNOW.EQ.1) THEN
55       PRINT *, 'WILL YOU ENTER KNOWN COVARIANCE MATRIX, OF '
         PRINT *, 'CONTROLS, MANUALLY OR BY DATAFILE (M or D): '
         READ *, ANSWER
         IF ((ANSWER.EQ.'D').OR.(ANSWER.EQ.'d')) THEN
            PRINT *, 'ENTER NAME OF DATAFILE CONTAINING KNOWN '
            PRINT *, 'COVARIANCE MATRIX OF CONTROLS: '
            READ *, COVFILE
            OPEN(UNIT=15, FILE=COVFILE, STATUS='OLD')
            DO 60 I=1,NX
               READ(15,*) (COVCV(I,J),J=1,NX)
60          CONTINUE
            CLOSE(15)
         ELSE
            IF ((ANSWER.NE.'M').AND.(ANSWER.NE.'m')) THEN
               PRINT *, 'INVALID RESPONSE, TRY AGAIN.'
               GOTO 55
            ENDIF
            PRINT *, '   INPUT THE REQUESTED VALUES: '
            PRINT *, ''
            DO 65 I=1,NX
               DO 65 J=1,NX
                  PRINT *, '      COVCV(',I,',',J,') = '
                  READ *, COVCV(I,J)
65          CONTINUE
         ENDIF
      ENDIF
70    PRINT *, 'WILL YOU ENTER THE [CONTROLS:RESPONSES] DATA '
      PRINT *, 'MANUALLY OR BY DATAFILE (M or D): '
      READ *, ANSWER
```

```
        IF ((ANSWER.EQ.'D').OR.(ANSWER.EQ.'d')) THEN
           PRINT *, 'ENTER NAME OF [CONTROLS|RESPONSES] DATAFILE: '
           READ *, XFILE
           OPEN(UNIT=20, FILE=XFILE, STATUS='OLD')
        ELSE
           IF ((ANSWER.NE.'M').AND.(ANSWER.NE.'m')) THEN
              PRINT *, 'INVALID RESPONSE, TRY AGAIN.'
              GOTO 70
           ENDIF
           PRINT *, '   MATRIX VALUES WILL BE REQUESTED AS REQUIRED'
        ENDIF
 75     PRINT *, 'DO YOU WANT YOUR INPUTS SENT TO A DATAFILE FOR '
        PRINT *, 'FUTURE USE (Y/N): '
        READ *, ANSWER
        IF ((ANSWER.EQ.'Y').OR.(ANSWER.EQ.'y')) THEN
           PRINT *, '   ENTER NAME OF INPUT DATAFILE TO CREATE: '
           READ *, OUTFILE
           OPEN(UNIT=25, FILE=OUTFILE)
           WRITE(25,*) NX,NY,KEEPERS,NUMREPS,META
           WRITE(25,*) IKNOW,IWRITE,METHOD
           WRITE(25,*) SIG
           WRITE(25,*) (VECMUC(I),I=1,NX)
           WRITE(25,*) (VECCBAR(I),I=1,NX)
           WRITE(25,*) (VECMUY(I),I=1,NY)
           WRITE(25,*) (VECYBAR(I),I=1,NY)
           WRITE(25,*) TITLE
           WRITE(25,*) (CONTROL(I),I=1,NX)
           WRITE(25,*) (RESPONS(I),I=1,NY)
           IF (IKNOW.EQ.1) THEN
              WRITE(25,*) COVFILE
              IF (COVFILE.EQ.OUTFILE) THEN
                 DO 80 I=1,NX
                    WRITE(25,*) (COVCV(I,J),J=1,NX)
 80              CONTINUE
              ENDIF
           ENDIF
           WRITE(25,*) XFILE
           CLOSE(25)
        ELSE
           IF ((ANSWER.NE.'N').AND.(ANSWER.NE.'n')) THEN
              PRINT *, '   INVALID INPUT, TRY AGAIN.'
              GOTO 75
           ENDIF
        ENDIF
      ENDIF
C
      NVAR = NX + NY
      KNX = 2**NX
C
C *******************************************
C *  TEST PRIMARY VARIABLES INPUT           *
C *******************************************
```

```
C
      I=0
      PRINT *, ''
      IF (NX.GT.Z1) THEN
         PRINT 1575, Z1
         I=I+1
      ENDIF
      IF (NY.GT.Z2) THEN
         PRINT 1580, Z2
         I=I+1
      ENDIF
      IF (KEEPERS.GT.Z4) THEN
         PRINT 1585
         PRINT 1586, Z4
         I=I+1
      ENDIF
      IF (NUMREPS.GT.Z6) THEN
         PRINT 1590
         PRINT 1591, Z6
         I=I+1
      ENDIF
      IF (META.GT.Z7) THEN
         PRINT 1595, Z7
         I=I+1
      ENDIF
      IF (I.GT.0) THEN
         PRINT 1600, I
         STOP
      ENDIF
C
C ********************************************
C *  INITIAL DATA OUTPUT                     *
C ********************************************
C
      PRINT *, 'ENTER NAME OF FILE FOR PROGRAM OUTPUT: '
      READ *, OUTFILE
      PRINT *, ''
      OPEN(UNIT=30, FILE=OUTFILE, STATUS='NEW')
      WRITE(30,1500) TITLE,META,NUMREPS,META*NUMREPS
      WRITE(30,1515)
      WRITE(30,1505) META*NUMREPS
      DO 90 I=1,NY
         WRITE(30,1510) I,RESPONS(I),VECYBAR(I),VECMUY(I)
   90 CONTINUE
      WRITE(30,1515)
      WRITE(30,1520) META*NUMREPS
      DO 95 I=1,NX
         WRITE(30,1510) I,CONTROL(I),VECCBAR(I),VECMUC(I)
   95 CONTINUE
      WRITE(30,1515)
C
```

```
C ***************************************
C * DECLARE IF COVARIANCE MATRIX        *
C * OF CONTROLS IS KNOWN OR ESTIMATED   *
C *                                     *
C *   IKNOW = 0, COV MATRIX ESTIMATED   *
C *   IKNOW = 1, KNOWN COV MATRIX USED  *
C ***************************************
C
      IF (IKNOW.EQ.0) THEN
         WRITE(30,1525)
      ELSE
         WRITE(30,1530)
      ENDIF
      WRITE(30,1515)
C
C ***************************************
C *  PROVIDE HEADING FOR REMAINDER OF   *
C *  PROGRAM OUTPUT                     *
C ***************************************
C
      WRITE(30,1515)
      WRITE(30,1535)
      WRITE(30,1515)
C
C **********************************
C * DEFINE INPUT VECTOR FOR IMSL   *
C * SUBROUTINE 'BECOVM'            *
C **********************************
C
      NBR(1)=Z3
      NBR(2)=NUMREPS
      NBR(3)=NUMREPS
      NBR(4)=1
      NBR(5)=1
      NBR(6)=0
      IX=Z6
C
C ****************************************************************
C * BEGIN MAIN PROGRAM                                           *
C ****************************************************************
C
C ********************************************
C * MAKE THE F TABLE                         *
C ********************************************
C
       IP=NY
       KK=NUMREPS
       CALL FTABL(FF,NX,Z1)
       PRINT *,'F TABLE:',(FF(I),I=0,NX)
       PRINT *, ''
C
```

```
C *******************************************
C * INITIALIZE COVERAGE AND VOLUME          *
C * REDUCTION ACCUMULATORS                  *
C *******************************************
C
      DO 100 IZ=1,4
         ICTOT(IZ)=0
 100  CONTINUE
C
      DO 105 IZ=1,2
         SUMDEV(IZ)=0.
         SUMVU(IZ)=0.
         VR(IZ)=0.
 105  CONTINUE
C
C ************************************************
C * THIS IS THE META EXPERIMENT LOOP             *
C ************************************************
C
      DO 1000 MM=1,META
C
C ********************************
C * INITIALIZE ARRAYS            *
C ********************************
C
         NUMREG=0
         DO 110 IZ=1,KEEPERS
            DO 110 JZ=1,NX
               DO 110 KZ=1,2
                  REGR(IZ,JZ,KZ)=0.
 110     CONTINUE
C
         DO 115 IZ=1,KNX
            DO 115 JZ=1,NX
               MODELS(IZ,JZ)=0
 115     CONTINUE
C
         DO 120 IZ=1,NVAR
            DO 120 JZ=1,NVAR
               DO 120 KZ=1,NVAR
                  A(IZ,JZ,KZ)=0.
 120     CONTINUE
C
         DO 125 IZ=1,KEEPERS
            BUFF(IZ)=0
            BUFF2(IZ)=0
 125     CONTINUE
C
         DO 130 IZ=1,NX
            IBUFF(IZ)=0
 130     CONTINUE
C
```

```
C ******************************************
C * READ THE DATA                          *
C *   (EACH RECORD => [CONTROLS:RESPONSES]) *
C * COMPUTE THE COVARIANCE MATRIX          *
C * SAVE SAMPLE MEANS                      *
C * BOUND THE GENERALIZED VARIANCE         *
C ******************************************
C
        IF ((ANSWER.EQ.'M').OR.(ANSWER.EQ.'m')) THEN
           PRINT *, 'ENTER ELEMENTS OF [CONTROL:RESPONSE] MATRIX'
           PRINT *, ''
           DO 131 I=1,NUMREPS
              DO 132 J=1,NVAR
                    PRINT *, '   X(',I,',',J,') = '
                    READ *, X(I,J)
 132          CONTINUE
 131       CONTINUE
        ELSE
           DO 135 I=1,NUMREPS
              READ(20,*)(X(I,J),J=1,NVAR)
 135       CONTINUE
        ENDIF
C
        CALL BECOVM(X,IX,NBR,TEMP,XM,VCV,IER)
C
        DO 140 I=1,NX
           CBAR(I)=XM(I)
 140    CONTINUE
C
        DO 145 I=NX+1,NVAR
           YBAR(I-NX)=XM(I)
 145    CONTINUE
C
        CALL VCVTSF(VCV,Z3,FULL,Z3)
C
        DO 150 I=1,NVAR
           DO 150 J=1,NVAR
              A(1,I,J)=FULL(I,J)
 150    CONTINUE
C
        IS=1
        DO 155 II=1,NY
           DO 155 JJ=1,NY
              IF (JJ.GE.II) THEN
                 RSS(II,JJ)=A(IS,NX+II,NX+JJ)
                 RSS(JJ,II)=RSS(II,JJ)
              ENDIF
 155    CONTINUE
C
        CALL LINV3F(RSS,DUM,4,NY,Z2,D1,D2,WKAREA,IER)
        IF (IER.NE.0) PRINT *,'I DIED BELOW 155 (MAIN)'
        DET=D1*2**D2
```

```
      BIG=(FLOAT(NUMREPS-1)/FLOAT(NUMREPS-NX-2))**NY
      TWO=2*BIG*D1*2**D2
C
C *****************************************
C * STUFF THE BOOKKEEPING ARRAY WITH THE    *
C * BOUND                                   *
C *****************************************
C
      DO 160 II=1,KEEPERS
         DO 160 JJ=1,NX
            REGR(II,JJ,1)=TWO
 160  CONTINUE
C
C ********************************************
C *  THE FOLLOWING SECTION IS PERFORMED IF THE   *
C *  METHOD OF ENUMERATED SUBSETS IS SELECTED    *
C *  (i.e. METHOD = 0)                           *
C ********************************************
C
      IF (METHOD.EQ.0) THEN
C
C *****************************************
C *  CONDUCT BINARY SEARCH OF THE           *
C *  REGRESSION TREE:                       *
C *     FURNIVAL AND WILSON (1974)          *
C *****************************************
C
         K=NX
C
         DO 165 L=1,K
            NK(L)=0
 165     CONTINUE
C
         NK(K+1)=1
         L=1
 170     NK(L)=1
C
         DO 175 M=L,K
            IF (NK(M+1).EQ.1) GOTO 180
 175     CONTINUE
C
 180     IB=K-M+1
         IS=K-L+2
         IP=K-L+1
         KP=NVAR
         CALL GAUSS(IB,IS,IP,A,KP,MAX,Z3)
C
C *****************************************
C * CALCULATION OF THE GENERALIZED RESIDUAL *
C * COVARIANCE                              *
C *****************************************
C
```

```
          DO 185 II=1,NY
             DO 185 JJ=1,NY
                IF (JJ.GE.II) THEN
                   RSS(II,JJ)=A(IS,NX+II,NX+JJ)
                   RSS(JJ,II)=RSS(II,JJ)
                ENDIF
 185      CONTINUE
C
          IF (IKNOW.EQ.0) THEN
             CALL LINV3F(RSS,DUM,4,NY,Z2,D1,D2,WKAREA,IER)
             IF (IER.NE.0) PRINT *,'I DIED BELOW 185'
             DET=D1*2**D2
          ENDIF
C
C *******************************************
C * BOOKKEEPING LOGIC TO SAVE M=KEEPERS      *
C * BEST REGRESSIONS OF ALL J SUBSETS SIZES *
C *******************************************
C
          MV=0
          DO 190 N=1,NX
             MV=MV+NK(N)
 190      CONTINUE
C
          IF (IKNOW.EQ.0) THEN
             CONST=(FLOAT(NUMREPS-1)/FLOAT(NUMREPS-MV-1))
             DET=DET*CONST**NY
          ELSE
             CALL COVKNOW(RSS,NY,Z2,FULL,NVAR,Z3,TARGET,DUM,
     &                    NUMREPS,MV,DET)
          ENDIF
C
          DO 195 J=1,KEEPERS
             IF (DET.LT.REGR(J,MV,1)) THEN
                NUMREG=NUMREG+1
                DO 200 JJ=J,KEEPERS-1
                   BUFF(JJ+1)=REGR(JJ,MV,1)
                   BUFF2(JJ+1)=REGR(JJ,MV,2)
 200            CONTINUE
                REGR(J,MV,1)=DET
                REGR(J,MV,2)=NUMREG
                DO 205 JJ=J+1,KEEPERS
                   REGR(JJ,MV,1)=BUFF(JJ)
                   REGR(JJ,MV,2)=BUFF2(JJ)
 205            CONTINUE
                CALL KEEPIT(NUMREG,NK,NX,MODELS,Z1,Z3,Z5)
                GOTO 210
             ENDIF
 195      CONTINUE
 210      CONTINUE
C
```

```
            DO 215 L=1,K
               IF (NK(L).EQ.0) GOTO 170
               NK(L)=0
 215        CONTINUE
         ENDIF
C
C ************************************************
C *  ENUMERATED SUBSETS CODE ENDS ON ABOVE LINE  *
C ************************************************
C
C ************************************************
C *  THE FOLLOWING SECTION IS PERFORMED IF THE   *
C *  STEPWISE PROCEDURE [FORWARD SELECTION] IS   *
C *  SELECTED (i.e. METHOD = 1)                  *
C ************************************************
C
         IF (METHOD.EQ.1) THEN
C
C *******************************************
C *  CONDUCT STEPWISE (FORWARD SELECTION)   *
C *  SEARCH OF THE REGRESSION TREE.         *
C *  THIS IS A MODIFIED VERSION OF THE      *
C *  NATURAL SEARCH PROCEDURE WRITTEN BY:   *
C *     FURNIVAL AND WILSON (1974)          *
C *******************************************
C
            K=NX
C
            DO 220 L=1,K+1
               IND1(L)=0
               IND2(L)=0
               TIND(L)=0
 220        CONTINUE
C
            M=K
            IB=0
            IS=1
            TMV=1
C
 225        IB=MOD(IB,MAX)+1
C
            DO 230 L=M,K
               IF (IND2(L).LT.L) GOTO 230
               IND2(L-1)=IND2(L-1)+1
               IND2(L)=IND2(L-1)
 230        CONTINUE
C
 235        IND2(K)=IND2(K)+1
            IS=MOD(IS,MAX)+1
C
            MV=0
```

```
      DO 240 I1=1,K
         IF (IND2(I1).GT.0) MV=MV+1
 240  CONTINUE
C
      IF (MV.GT.TMV) THEN
         TMV=TMV+1
         DO 245 I1=1,K
            IND1(I1)=TIND(I1)
 245     CONTINUE
      ENDIF
C
      IF (MV.EQ.1) GOTO 260
C
      DO 250 I1=1,K
         FLAG=0
         DO 255 I2=1,K+1
            IF (IND2(I2).EQ.IND1(I1)) FLAG=1
 255     CONTINUE
         IF (FLAG.EQ.0) GOTO 295
 250  CONTINUE
C
 260  CONTINUE
      IP=IND2(K)
      KP=NVAR
      IB2=1
      IS2=2
      IF (MV.GT.2) THEN
         DO 261 I1=1,NX
            IF (IND2(I1).GT.0) THEN
               IF (IS2.EQ.2) THEN
                  IS2=3
               ELSE
                  IS2=2
               ENDIF
               IP=IND2(I1)
               CALL GAUSS(IB2,IS2,IP,A,KP,MAX,Z3)
               IB2=IS2
            ENDIF
 261     CONTINUE
      ELSE
         CALL GAUSS(IB,IS,IP,A,KP,MAX,Z3)
         IS2=IS
      ENDIF
C
C ********************************************
C * CALCULATION OF THE GENERALIZED RESIDUAL *
C * COVARIANCE                               *
C ********************************************
C
```

```
            DO 265 II=1,NY
               DO 265 JJ=1,NY
                  IF (JJ.GE.II) THEN
                     RSS(II,JJ)=A(IS2,NX+II,NX+JJ)
                     RSS(JJ,II)=RSS(II,JJ)
                  ENDIF
 265        CONTINUE
C
            IF (IKNOW.EQ.0) THEN
               CALL LINV3F(RSS,DUM,4,NY,Z2,D1,D2,WKAREA,IER)
               IF (IER.NE.0) PRINT *,'I DIED BELOW 265'
               DET=D1*2**D2
            ENDIF
C
C ******************************************
C * BOOKKEEPING LOGIC TO SAVE M=KEEPERS     *
C * BEST REGRESSIONS OF ALL J SUBSETS SIZES *
C ******************************************
C
            IF (IKNOW.EQ.0) THEN
               CONST=(FLOAT(NUMREPS-1)/FLOAT(NUMREPS-MV-1))
               DET=DET*CONST**NY
            ELSE
               CALL COVKNOW(RSS,NY,Z2,FULL,NVAR,Z3,TARGET,DUM,
     &                      NUMREPS,MV,DET)
            ENDIF
C
            DO 266 I1=1,NX
               NK(I1)=0
 266        CONTINUE
            NK(NX+1)=1
C
            DO 270 J=1,KEEPERS
               IF (DET.LT.REGR(J,MV,1)) THEN
                  NUMREG=NUMREG+1
                  DO 275 JJ=J,KEEPERS-1
                     BUFF(JJ+1)=REGR(JJ,MV,1)
                     BUFF2(JJ+1)=REGR(JJ,MV,2)
 275              CONTINUE
                  REGR(J,MV,1)=DET
                  REGR(J,MV,2)=NUMREG
                  DO 280 JJ=J+1,KEEPERS
                     REGR(JJ,MV,1)=BUFF(JJ)
                     REGR(JJ,MV,2)=BUFF2(JJ)
 280              CONTINUE
                  DO 285 I1=1,NX
                     TIND(I1)=IND2(I1)
                     IF (IND2(I1).GT.0) THEN
                        NK(NX+1-IND2(I1))=1
                     ENDIF
 285              CONTINUE
                  CALL KEEPIT(NUMREG,NK,NX,MODELS,Z1,Z3,Z5)
```

```
                  GOTO 290
               ENDIF
 270        CONTINUE
C
 290        CONTINUE
C
 295        CONTINUE
            IF (IND2(K).LT.K) GOTO 235
            IS=IS-1
            IF (IND2(M).EQ.M) M=M-1
            IF (M.GT.0) GOTO 225
         ENDIF
C
C ************************************************
C *  STEPWISE (FORWARD SELECTION) CODE ENDS ON   *
C *  ABOVE LINE                                  *
C ************************************************
C
C *******************************************
C * THIS BLOCK IS FOR BEST M SUBSETS MODE   *
C * OF OPERATION                            *
C *******************************************
C
         IF (IWRITE.EQ.1) THEN
            DO 300 I=1,NX
               WRITE(30,1540) KEEPERS,I
               DO 300 J=1,KEEPERS
                  IVAR=0
                  IIN=0
                  DO 305 II=NX,1,-1
                     IVAR=IVAR+1
                     IF (IFIX(REGR(J,I,2)+.0001).EQ.0) GOTO 300
                     IF (MODELS(IFIX(REGR(J,I,2)+.0001),II).EQ.1) THEN
                        IIN=IIN+1
                        IBUFF(IIN)=IVAR
                     ENDIF
 305              CONTINUE
                  RDET=REGR(J,I,1)
                  WRITE(30,1545) MM,RDET,(IBUFF(IJ),IJ=1,IIN)
 300        CONTINUE
         ENDIF
C
C *******************************************
C * FOR EACH SUBSET COMPUTE THE CRITERION   *
C * AND SAVE THE MINIMUM                    *
C *******************************************
C
         IF (IWRITE.EQ.0) THEN
            IP=NY
            KK=NUMREPS
            DO 310 IQ=1,NX
```

```
               IF (IKNOW.EQ.0) THEN
                  REGR(1,IQ,1)=REGR(1,IQ,1)*C3(KK,IQ,IP)*
     &            CFRONT(KK,IQ,IP)*FF(IQ)*C5(KK,IQ,IP)
               ELSE
                  REGR(1,IQ,1)=REGR(1,IQ,1)*C4(KK,IQ,IP)*
     &            CFRONT(KK,IQ,IP)*FF(IQ)*C5(KK,IQ,IP)
               ENDIF
               IF (IQ.EQ.1) RMIN=REGR(1,IQ,1)+1000.
 310        CONTINUE
            DO 315 IQ=1,NX
               IF (REGR(1,IQ,1).LT.RMIN) THEN
                  RMIN=REGR(1,IQ,1)
                  IAT= REGR(1,IQ,2)
               ENDIF
 315        CONTINUE
            IVAR=0
            IIN=0
            DO 320 II=NX,1,-1
               IVAR=IVAR+1
               IF (MODELS(IAT,II).EQ.1) THEN
                  IIN=IIN+1
                  IBUFF(IIN)=IVAR
               ENDIF
 320        CONTINUE
            SP=RMIN
            WRITE(30,1545) MM,SP,(IBUFF(IJ),IJ=1,IIN)
C
C *******************************************
C * FIND THE VOLUME REDUCTION AND INDICATE  *
C * COVERAGE                                *
C *******************************************
C
            CALL COVER(VCV,MODELS,KNX,NX,NVAR,IAT,IIN,YBAR,CBAR,
     &                VECMUC,NY,VECMUY,NUMREPS,FF,IH,ICOVER,VOLRED,
     &                VECYBAR,IKNOW,COVCV,VU,DIFF)
C
C *******************************************
C * COVERAGE AND VOLUME REDUCTION TALLYS    *
C *******************************************
C
            DO 325 IC=1,4
               ICTOT(IC)=ICTOT(IC)+ICOVER(IC)
 325        CONTINUE
            DO 330 IC=1,2
              SUMDEV(IC)=SUMDEV(IC)+DIFF
              SUMVU(IC)=SUMVU(IC)+VU
 330        CONTINUE
         ENDIF
         PRINT *,'THIS IS META-EXPERIMENT #',MM,'ICOVER ',ICOVER
1000  CONTINUE
C
```

```
C ***********************************************
C * META EXPERIMENT LOOP ENDS ON ABOVE LINE      *
C ***********************************************
C
      DO 1005 IZ=1,2
         SUMDEV(IZ)=SUMDEV(IZ)/FLOAT(META)
         SUMVU(IZ)=SUMVU(IZ)/FLOAT(META)
         VR(IZ)=SUMDEV(IZ)/SUMVU(IZ)
1005  CONTINUE
C
      DO 1010 IZ=1,4
         COVERAG(IZ)=FLOAT(ICTOT(IZ))/FLOAT(META)
1010  CONTINUE
C
      WRITE(30,1515)
      WRITE(30,1550) COVERAG(1)
      WRITE(30,1551) VR(1)
      WRITE(30,1555) COVERAG(2)
      WRITE(30,1515)
      WRITE(30,1560) COVERAG(3)
      WRITE(30,1561) VR(1)
      WRITE(30,1565) COVERAG(4)
C
      CLOSE(5)
      CLOSE(10)
      CLOSE(30)
      STOP
C
C ******************************************
C *  FORMAT STATEMENTS (MAIN PROGRAM)      *
C ******************************************
C
1500  FORMAT(1X,A25,'META = ',I3,', NUMREPS = ',I3,', TOTAL REPS = ',
     &I4)
1505  FORMAT(1X,'THE RESPONSE ARE',13X,'MEAN ',I4,' REPS',2X,
     &'STEADY STATE MEAN'/)
1510  FORMAT(2X,I2,1X,A25,F12.5,4X,F12.5)
1515  FORMAT(' ')
1520  FORMAT(1X,'THE CANDIDATE CONTROLS ARE',3X,'MEAN ',I4,' REPS',
     &2X,'STEADY STATE MEAN'/)
1525  FORMAT(/,1X,'COVARIANCE MATRIX OF CONTROLS WAS ESTIMATED')
1530  FORMAT(/,1X,'KNOWN COVARIANCE MATRIX OF CONTROLS WAS USED')
1535  FORMAT(1X,'META#',3X,'  CRITERION   ',10X,'VARIABLE SUBSET')
1540  FORMAT(10X,'BEST ',I2,' REGRESSIONS WITH ',I2,' VARIABLES'//)
1545  FORMAT(1X,I4,2X,E16.8,10X,30(I2,1X))
1550  FORMAT(1X,'CONTRLD COVERAGE ON STEADY STATE MEANS   =',F12.8)
1551  FORMAT(1X,'                    AND VOLUME REDUCTION   =',E16.8)
1555  FORMAT(1X,'UNCONTRLD COVERAGE ON STEADY STATE MEANS =',F12.8)
1560  FORMAT(1X,'CONTRLD COVERAGE ON SAMPLE MEAN OF 1000 REPS  =',F12.8)
1561  FORMAT(1X,'                          AND VOLUME REDUCTION  =',E16.8)
1565  FORMAT(1X,'UNCONTRLD COVERAGE ON SAMPLE MEAN OF 1000 REPS=',F12.8)
1575  FORMAT(1X,'# OF CANDIDATE CONTROLS EXCEED PROGRAM LIMIT OF',I3)
```

```
1580  FORMAT(1X,'* OF RESPONSES EXCEED PROGRAM LIMIT OF',I3)
1585  FORMAT(1X,'* OF BEST REGRESSIONS TO KEEP EXCEED PROGRAM')
1586  FORMAT(1X,'LIMIT OF',I3)
1590  FORMAT(1X,'* OF REPLICATIONS PER META EXPERIMENT EXCEED')
1591  FORMAT(1X,'PROGRAM LIMIT OF',I3)
1595  FORMAT(1X,'* OF META EXPERIMENTS EXCEED PROGRAM LIMIT OF',I4)
1600  FORMAT(1X,'PROGRAM LIMITS EXCEEDED BY ',I1,' PRIMARY INPUTS')
      END
C
C ***************************************************************
C * END MAIN PROGRAM                                            *
C ***************************************************************
C
C ***************************************************************
C * SUBROUTINES                                                 *
C ***************************************************************
C
C ***************************************************
C *  SUBROUTINF COVER                               *
C *                                                 *
C *  THIS SUBROUTINE DOES THE COVERAGE AND VOLUME   *
C *  REDUCTION CALCULATIONS FOR THE OPTIMAL CONTROL *
C *  SUBSET                                         *
C ***************************************************
C
      SUBROUTINE COVER(VCV,MODELS,KNX,NX,NVAR,IAT,IIN,YBAR,CBAR
     &,VECMUC,NY,VECMUY,NUMREPS,FF,IH,ICOVER,VOLRED,VECYBAR,IKNOW
     &,COVCV,VU,DIFF)
C
      INTEGER Z1, Z2, Z3, Z5, Z8
      REAL PI
C
      PARAMETER (Z1=8, Z2=8, Z3=Z1+Z2, Z5=2**Z1, Z8=((Z3*(Z3+1))/2))
      PARAMETER (PI=3.1415927)
C
      REAL VCV(Z8),YBAR(Z2),CBAR(Z1),VECMUC(Z1),VECMUY(Z2)
     &,FF(0:Z1),VECYBAR(Z2),VOLRED(2),COVCV(Z1,Z1)
C
      INTEGER MODELS(Z5,Z1),IH(Z3),ICOVER(4),IIN
C
      REAL SCBAR(Z1),SVECMU(Z1),SUBV(Z8),SUBVF(Z3,Z3),B(Z3)
     &,WKAREA(2*Z3),BUFF1(Z3,Z3),BUFF2(Z2,Z1),BETA(Z2,Z1)
     &,CDEV1(1,Z1),CDEV2(Z1,1),EXPL(Z2,Z2),DEV(Z2,1),YBHAT(Z2)
     &,BUFF3(Z1,Z2),BUFF4(Z2,Z2),SYDOTC(Z2,Z2),HPH(1,1),T1(1,Z1)
     &,YMD1(1,Z2),YMD2(Z2,1),T2(1,Z2),OBS(1,1),BUFF5(Z2,Z2)
     &,BUFF6(Z2,Z2),YMD3(1,Z2),YMD4(Z2,1),OBS2(1,1)
     &,SYMCOVC((Z1*(Z1+1))/2),SUBCOVC((Z1*(Z1+1))/2)
     &,FULCOVC(Z1,Z1),GAMM(Z2,Z1),EHAT(Z2,Z2),BUFF9(Z2,Z2)
     &,CANCORR(Z2,Z2),REIGS(Z2),EIGS(2*Z2),DUMMY(Z2,Z2)
     &,WK(Z2)
C
```

```
      INTEGER IH2(Z1)
C
      COMPLEX CEIGS(Z2)
C
      EQUIVALENCE (EIGS(1),CEIGS(1))
C
C ******************************************
C * INITIALIZE COVERAGE AND VOLUME         *
C * REDUCTION VECTORS                      *
C ******************************************
C
      DO 10 I=1,4
         ICOVER(I)=0
   10 CONTINUE
C
      DO 15 I=1,2
         VOLRED(I)=0.
   15 CONTINUE
C
C ******************************************
C * FIND THE SUBMATRIX FOR THE SELECTED    *
C * MODEL                                  *
C ******************************************
C
      DO 20 I=1,NVAR
         IF (I.LE.NX) THEN
            IH(I)=0
            IH2(I)=0
         ELSE
            IH(I)=1
         ENDIF
   20 CONTINUE
C
      IVAR=0
      DO 25 II=NX,1,-1
         IVAR=IVAR+1
         IF (MODELS(IAT,II).EQ.1) THEN
            IH(IVAR)=1
            IH2(IVAR)=1
         ENDIF
   25 CONTINUE
C
      M1=Z3
      CALL RLSUBM(VCV,M1,IH,SUBV,M2)
C
C ******************************************
C * FIND THE SUBVECTOR (POPULATION AND     *
C * SAMPLE) OF THE CONTROL MEANS           *
C ******************************************
C
      INDEX=0
```

```
      DO 30 II=1,NX
         IF (IH(II).EQ.1) THEN
            INDEX=INDEX+1
            SCBAR(INDEX)=CBAR(II)
            SVECMU(INDEX)=VECMUC(II)
         ENDIF
  30  CONTINUE
C
C ****************************************
C * BUFFER THE COVARIANCE MATRIX OF      *
C * SELECTED CONTROLS AND RESPONSES      *
C ****************************************
C
      CALL VCVTSF(SUBV,M2,SUBVF,Z3)
C
      DO 35 I=1,M2
         DO 35 J=1,M2
            BUFF1(I,J)=SUBVF(I,J)
  35  CONTINUE
C
C ****************************************
C * INVERT THE COVARIANCE SUBMATRIX OF   *
C * CONTROLS                             *
C ****************************************
C
      CALL LINV3F(SUBVF,B,1,IIN,Z3,D1,D2,WKAREA,IER)
      IF (IER.NE.0) PRINT *,'I DIED BELOW 35 (SUBROUTINE COVER)'
C
C ****************************************
C * BUFFER THE CROSS-COVARIANCE SUBMATRICES *
C * OF SELECTED CONTROLS WITH RESPONSES  *
C ****************************************
C
      DO 40 I=IIN+1,M2
         DO 40 J=1,IIN
            BUFF2(I-IIN,J)=BUFF1(I,J)
            BUFF3(J,I-IIN)=BUFF1(I,J)
  40  CONTINUE
C
C ****************************************
C * BUFFER THE COVARIANCE SUBMATRIX OF   *
C * RESPONSES                            *
C ****************************************
C
      DO 45 I=IIN+1,M2
         DO 45 J=IIN+1,M2
            BUFF4(I-IIN,J-IIN)=BUFF1(I,J)
            BUFF6(I-IIN,J-IIN)=BUFF1(I,J)
  45  CONTINUE
C
```

```
C ********************************************
C * FIND THE BETA HAT MATRIX ( CONTROL       *
C * COEFFICIENTS ) OR THE GAMMA HAT MATRIX  *
C ********************************************
C
      IF (IKNOW.EQ.0) THEN
         CALL VMULFF(BUFF2,SUBVF,NY,IIN,IIN,Z2,Z3,BETA,Z2,IER)
      ELSE
         CALL VMULFF(BUFF2,SUBVF,NY,IIN,IIN,Z2,Z3,BETA,Z2,IER)
         CALL VCVTFS(COVCV,NX,Z1,SYMCOVC)
         CALL RLSUBM(SYMCOVC,NX,IH2,SUBCOVC,IORDER)
         CALL VCVTSF(SUBCOVC,IORDER,FULCOVC,Z1)
         CALL LINV3F(FULCOVC,B,1,IIN,Z1,D1,D2,WKAREA,IER)
         IF (IER.NE.0) PRINT *,'I DIED BELOW 45 (SUBROUTINE COVER)'
         CALL VMULFF(BUFF2,FULCOVC,NY,IIN,IIN,Z2,Z1,GAMM,Z2,IER)
      ENDIF
C
C ********************************************
C * FIND THE VECTOR OF CORRECTIONS TO       *
C * CONTROL Y BAR                           *
C ********************************************
C
      DO 50 I=1,IIN
         CDEV1(1,I)=SCBAR(I)-SVECMU(I)
         CDEV2(I,1)=CDEV1(1,I)
   50 CONTINUE
C
      IF (IKNOW.EQ.0) THEN
         CALL VMULFF(BETA,CDEV2,NY,IIN,1,Z2,Z1,DEV,Z2,IER)
      ELSE
         CALL VMULFF(GAMM,CDEV2,NY,IIN,1,Z2,Z1,DEV,Z2,IER)
      ENDIF
C
C ********************************************
C * FIND THE CONTROLLED ESTIMATOR OF THE    *
C * MEAN                                    *
C ********************************************
C
      DO 55 I=1,NY
         YBHAT(I)=YBAR(I)-DEV(I,1)
   55 CONTINUE
C
C ********************************************
C * FIND THE MATRIX OF EXPLAINED COVARIANCE *
C * DUE TO CONTROL                          *
C ********************************************
C
      CALL VMULFF(BETA,BUFF3,NY,IIN,NY,Z2,Z1,EXPL,Z2,IER)
C
C ********************************************
C * FIND THE RESIDUAL COVARIANCE            *
C ********************************************
```

```
C
      C1=(FLOAT(NUMREPS-1)/FLOAT(NUMREPS-IIN-1))
C
      DO 60 I=1,NY
         DO 60 J=1,NY
            SYDOTC(I,J)=(BUFF4(I,J)-EXPL(I,J))*C1
            BUFF5(I,J) =SYDOTC(I,J)
  60  CONTINUE
C
C ******************************************
C * FIND THE ESTIMATOR SIGMA TILDE HAT     *
C ******************************************
C
      IF (IKNOW.EQ.1) THEN
         CONST1=(FLOAT(NUMREPS-2))/(FLOAT(NUMREPS*(NUMREPS-1)))
         CONST2=(FLOAT(IIN+1))/(FLOAT(NUMREPS*(NUMREPS-1)))
         DO 65 I=1,NY
            DO 65 J=1,NY
               EHAT(I,J)=(CONST1*SYDOTC(I,J))+(CONST2*BUFF4(I,J))
               BUFF9(I,J)=EHAT(I,J)
  65     CONTINUE
      ENDIF
C
C ******************************************
C * FIND THE INVERSE RESIDUAL COVARIANCE   *
C * MATRIX                                 *
C ******************************************
C
      IF (IKNOW.EQ.0) THEN
         CALL LINV3F(SYDOTC,B,1,NY,Z2,D1,D2,WKAREA,IER)
         IF (IER.NE.0) PRINT *,'I DIED BELOW 65 [1] (SUBR COVER)'
      ELSE
         CALL LINV3F(EHAT,B,1,NY,Z2,D1,D2,WKAREA,IER)
         IF (IER.NE.0) PRINT *,'I DIED BELOW 65 [1] (SUBR COVER)'
      ENDIF
C
C ******************************************
C * COMPUTE THE DEVIATIONS FROM THE        *
C * STEADY-STATE RESPONSE VECTOR           *
C * (BOTH CASES: CONTROLLED/UNCONTROLLED)  *
C ******************************************
C
      DO 70 I=1,NY
         YMD1(1,I)=YBHAT(I)-VECMUY(I)
         YMD2(I,1)=YMD1(1,I)
         YMD3(1,I)=YBAR(I)-VECMUY(I)
         YMD4(I,1)=YMD3(1,I)
  70  CONTINUE
C
```

```
C ********************************************
C * COMPUTE H'H                              *
C * (NOTATION AS PER VENXATRAMAN AND         *
C * WILSON 1986)                             *
C ********************************************
C
      IF (IKNOW.EQ.0) THEN
         CALL VMULFF(CDEV1,SUBVF,1,IIN,IIN,1,Z3,T1,1.IER)
         CALL VMULFF(T1,CDEV2,1,IIN,1,1,Z1,HPH,1,IER)
      ENDIF
C
      IF (IKNOW.EQ.0) THEN
         X=(1./FLOAT(NUMREPS))+(1./FLOAT(NUMREPS-1))*HPH(1,1)
      ELSE
         X=1.
      ENDIF
C
C ********************************************
C * COMPUTE THE RIGHT HAND SIDE FOR THE      *
C * CONFIDENCE REGION AS PER RAO (1967)      *
C ********************************************
C
      C2=(FLOAT((NUMREPS-IIN-1)*NY)/FLOAT(NUMREPS-IIN-NY))
      F=EXP((1./FLOAT(NY))*ALOG(FF(IIN)))
      RHS=X*C2*F
C
C ********************************************
C * COMPUTE THE T**2 STATISTIC FOR THE CASE *
C * WHERE  CONTROLS ARE USED (STEADY STATE  *
C * ASSUMED)                                 *
C ********************************************
C
      IF (IKNOW.EQ.0) THEN
         CALL VMULFF(YMD1,SYDOTC,1,NY,NY,1,Z2,T2,1,IER)
         CALL VMULFF(T2,YMD2,1,NY,1,1,Z2,OBS,1,IER)
      ELSE
         CALL VMULFF(YMD1,EHAT,1,NY,NY,1,Z2,T2,1,IER)
         CALL VMULFF(T2,YMD2,1,NY,1,1,Z2,OBS,1,IER)
      ENDIF
C
C ********************************************
C * INDICATE COVERAGE FOR THE CASE WHERE     *
C * CONTROLS ARE USED (STEADY STATE ASSUMED)*
C ********************************************
C
      IF (OBS(1,1).LE.RHS) THEN
         ICOVER(1)=1
      ELSE
         ICOVER(1)=0
      ENDIF
C
```

```
C ********************************************
C * COMPUTE THE VOLUME REDUCTION             *
C ********************************************
C
      IF (IKNOW.EQ.0) THEN
         CALL LINV3F(BUFF4,B,4,NY,Z2,D1,D2,WKAREA,IER)
         IF (IER.NE.0) PRINT *,'I DIED BELOW 70 [1] (SUBR COVER)'
         UCDET=D1*2**D2
         CALL LINV3F(BUFF5,B,4,NY,Z2,D1,D2,WKAREA,IER)
         IF (IER.NE.0) PRINT *,'I DIED BELOW 70 [2] (SUBR COVER)'
         CDET=D1*2**D2
      ELSE
         CALL LINV3F(BUFF4,B,4,NY,Z2,D1,D2,WKAREA,IER)
         IF (IER.NE.0) PRINT *,'I DIED BELOW 70 [3] (SUBR COVER)'
         UCDET=D1*2**D2
         CALL LINV3F(BUFF9,B,4,NY,Z2,D1,D2,WKAREA,IER)
         IF (IER.NE.0) PRINT *,'I DIED BELOW 70 [4] (SUBR COVER)'
         CDET=D1*2**D2
      ENDIF
C
      TERM1=(CDET/UCDET)**(.5)*X**(FLOAT(NY)/2.)
      C3=FLOAT((NUMREPS-IIN-1)*(NUMREPS)*(NUMREPS-NY))
      C4=FLOAT((NUMREPS-IIN-NY)*(NUMREPS-1))
      TERM2=(C3/C4)**(FLOAT(NY)/2.)
      F2=EXP((1./FLOAT(NY))*ALOG(FF(0)))
      TERM3=(F/F2)**(FLOAT(NY)/2.)
      VOLRED(1)=(1.-(TERM1*TERM2*TERM3))*100.
C
C ********************************************
C * COMPUTE THE ACTUAL VOLUME OF THE         *
C * CONTROLLED ELLIPSOID                     *
C ********************************************
C
      POVER2=FLOAT(NY)/2.
      CC1=1./FLOAT(NY)
      CC2=(2.*PI**POVER2)/GAMMA(POVER2)
      CC3=FLOAT(NY*(NUMREPS-IIN-1))
      CC4=FLOAT(NUMREPS-IIN-NY)
      CC3=(CC3/CC4)**POVER2
      CC4=SQRT(FF(IIN))
      VC=CC1*CC2*CC3*CC4*SQRT(CDET)*(X**POVER2)
C
C ********************************************
C * COMPUTE THE ACTUAL VOLUME OF THE         *
C * UNCONTROLLED ELLIPSOID                   *
C ********************************************
C
      CC3=FLOAT(NY*(NUMREPS-1))
      CC4=FLOAT(NUMREPS*(NUMREPS-NY))
      CC3=CC3/CC4
      VU=CC1*CC2*SQRT(UCDET)*(CC3*FF(IIN))**POVER2
C
```

```
C *******************************************
C * COMPUTE THE DIFFERENCE (DIFF)           *
C *******************************************
C
      DIFF=VU-VC
C
C *******************************************
C * COMPUTE THE T**2 STATISTIC FOR THE CASE *
C * WHERE NO CONTROLS ARE USED              *
C *******************************************
C
      CALL LINV3F(BUFF6,B,1,NY,Z2,D1,D2,WKAREA,IER)
      IF (IER.NE.0) PRINT *,'I DIED BELOW 70 [5] (SUBR COVER)'
      CALL VMULFF(YMD3,BUFF6,1,NY,NY,1,Z2,T2,1,IER)
      CALL VMULFF(T2,YMD4,1,NY,1,1,Z2,OBS2,1,IER)
C
C *******************************************
C * COMPUTE THE RIGHT HAND SIDE FOR THE     *
C * CONFIDENCE REGION                       *
C *******************************************
C
      C5=(FLOAT((NUMREPS-1)*NY)/FLOAT((NUMREPS-NY)*NUMREPS))
      RHS2=EXP((1./FLOAT(NY))*ALOG(FF(0)))*C5
C
C *******************************************
C * INDICATE COVERAGE FOR THE CASE WHERE    *
C * NO CONTROLS ARE USED (STEADY STATE      *
C * ASSUMED)                                *
C *******************************************
C
      IF (OBS2(1,1).LE.RHS2) THEN
         ICOVER(2)=1
      ELSE
         ICOVER(2)=0
      ENDIF
C
C *******************************************
C * THE REMAINING ANALYSIS DUPLICATES THE   *
C * ABOVE, EXCEPT THAT THE GRAND MEAN OF    *
C * 1000 RESPONSES IS USED                  *
C *******************************************
C
C *******************************************
C * RECOMPUTE DEVIATIONS                    *
C *******************************************
C
      DO 75 I=1,NY
         YMD1(1,I)=YBHAT(I)-VECYBAR(I)
         YMD2(I,1)=YMD1(1,I)
         YMD3(1,I)=YBAR(I)-VECYBAR(I)
         YMD4(I,1)=YMD3(1,I)
   75 CONTINUE
```

```
C
C ******************************************
C * COMPUTE THE T**2 STATISTIC FOR THE     *
C * CASE WHERE CONTROLS ARE USED           *
C * (GRAND MEAN USED)                      *
C ******************************************
C
      IF (IKNOW.EQ.0) THEN
         CALL VMULFF(YMD1,SYDOTC,1,NY,NY,1,Z2,T2,1,IER)
         CALL VMULFF(T2,YMD2,1,NY,1,1,Z2,OBS,1,IER)
      ELSE
         CALL VMULFF(YMD1,EHAT,1,NY,NY,1,Z2,T2,1,IER)
         CALL VMULFF(T2,YMD2,1,NY,1,1,Z2,OBS,1,IER)
      ENDIF
C
C ******************************************
C * INDICATE COVERAGE FOR THE CASE WHERE   *
C * CONTROLS ARE USED                      *
C ******************************************
C
      IF (OBS(1,1).LE.RHS) THEN
         ICOVER(3)=1
      ELSE
         ICOVER(3)=0
      ENDIF
C
C ******************************************
C * COMPUTE THE T**2 STATISTIC, FOR THE    *
C * CASE WHERE NO CONTROLS ARE USED        *
C * (GRAND MEAN USED)                      *
C ******************************************
C
      CALL VMULFF(YMD3,BUFF6,1,NY,NY,1,Z2,T2,1,IER)
      CALL VMULFF(T2,YMD4,1,NY,1,1,Z2,OBS2,1,IER)
C
C ******************************************
C * INDICATE COVERAGE, FOR THE CASE WHERE  *
C * NO CONTROLS ARE USED                   *
C ******************************************
C
      IF (OBS2(1,1).LE.RHS2) THEN
         ICOVER(4)=1
      ELSE
         ICOVER(4)=0
      ENDIF
C
C ******************************************
C * THIS SECTION COMPUTES THE CANONICAL    *
C * CORRELATIONS FOR THE SUBSET MODELS AND *
C * THE FEASIBILTY BOUND FOR USING THE     *
C * KNOWN COVARIANCE MATRIX OF CONTROLS    *
C ******************************************
```

```
C
      IF (IKNOW.EQ.1) THEN
         CALL VMULFF(BUFF6,EXPL,NY,NY,NY,Z2,Z2,CANCORR,Z2,IER)
         CALL EIGRF(CANCORR,NY,Z2,0,EIGS,DUMMY,Z2,WK,IER)
C
         ICOUNT=0
         DO 80 I=1,NY
            DO 80 J=1,2
               ICOUNT=ICOUNT+1
               IF (J.EQ.1) REIGS(I)=SQRT(EIGS(ICOUNT))
  80     CONTINUE
C
         CTOP=FLOAT((NUMREPS+IIN-1)*(NUMREPS-IIN-2))/
     &      FLOAT((NUMREPS-1)*(NUMREPS-2))
         CBOT=CTOP*(FLOAT(NUMREPS-2)/FLOAT(NUMREPS+IIN-1))
         BOUND=SQRT((CTOP-1.)/(CBOT-1.))
         PRINT *,'CANONICAL CORRELATIONS ',REIGS,' BOUND ',BOUND
         PRINT *,EIGS
      ENDIF
C
      RETURN
      END
C
C ****************************************************
C *  SUBROUTINE COVKNOW                              *
C *                                                  *
C *  THIS SUBROUTINE RETURNS THE GENERALIZED VARIANCE *
C *  OF SIGMA TILDE HAT                              *
C ****************************************************
C
      SUBROUTINE COVKNOW(RSS,NY,Z2,FULL,NVAR,Z3,TARGET,DUM,NUMREPS,MV
     &                     ,DET)
C
      INTEGER Z2,Z3,NY,NVAR,NUMREPS
      REAL RSS(Z2,Z2),FULL(Z3,Z3),TARGET(Z2,Z2),DUM(Z2)
C
      C1=(FLOAT(NUMREPS-2)/FLOAT(NUMREPS*(NUMREPS-MV-1)))
      C2=(FLOAT(MV+1)/FLOAT(NUMREPS*(NUMREPS-1)))
      NX=NVAR-NY
C
      DO 10 I=1,NY
         DO 10 J=1,NY
            TARGET(I,J)=(C1*RSS(I,J))+(C2*FULL(NX+I,NX+J))
  10  CONTINUE
C
      CALL LINV3F(TARGET,DUM,4,NY,Z2,D1,D2,WKAREA,IER)
      IF (IER.NE.0) PRINT *,'I DIED BELOW 10 (SUBR COVKNOW)'
      DET=D1*2**D2
C
      RETURN
      END
C
```

```
C *****************************************************
C *  SUBROUTINE FTABL                                 *
C *                                                   *
C *  THIS SUBROUTINE COMPUTES AN F TABLE              *
C *  (TO THE POWER P)                                 *
C *****************************************************
C
      SUBROUTINE FTABL(FF,NX,Z1)
C
      COMMON /BLK1/ SIG,KK,IQQ,IP
C
      INTEGER Z1,KK,IQQ,IP,NSIG,NROOT,ITMAX
C
      REAL ROOT(1),LAST,FF(0:Z1),EPS,FP
C
      EXTERNAL F
C
      EPS=.001
      NSIG=5
      NROOT=1
      ITMAX=1000
      LAST=3.
C
      DO 10 IQQ=0,NX
         ROOT(1)=LAST
   15    CALL ZREAL2(F,EPS,EPS,EPS,NSIG,NROOT,ROOT,ITMAX,IER)
         IF (IER.EQ.33) THEN
            ROOT(1)=LAST+1.
            IER=0
            WRITE(6,1535)
            WRITE(6,*) ''
            GOTO 15
         ENDIF
         LAST=ROOT(1)
         FP=ROOT(1)**IP
         FF(IQQ)=FP
   10 CONTINUE
C
      RETURN
 1535 FORMAT(1X,'IGNORE LAST IER=33 WARNING --- REINITIALIZING')
      END
C
C *****************************************************
C *  SUBROUTINE GAUSS                                 *
C *                                                   *
C *  THIS SUBROUTINE PERFORMS THE PIVOTS FOR VARIABLE *
C *  INTRODUCTION INTO REGRESSION MODELS:             *
C *  FURNIVAL AND WILSON 1974                         *
C *****************************************************
C
      SUBROUTINE GAUSS(IB,IS,IP,A,KP,MAX,Z3)
C
```

```
      INTEGER IB,IS,IP,KP,MAX,Z3
C
      REAL A(MAX,Z3,Z3)
C
C ******************************************
C * TOLERANCE CHECK ON PIVOTS              *
C ******************************************
C
      LB=IP+1
      IF (A(IB,IP,IP).LT..01) THEN
         DO 10  L=LB,KP
            A(IS,IP,L)=A(IB,IP,L)
            DO 10 M=L,KP
               A(IS,L,M)=A(IB,L,M)
   10    CONTINUE
      ELSE
         DO 15 L=LB,KP
            A(IS,IP,L)=A(IB,IP,L)/A(IB,IP,IP)
            DO 15 M=L,KP
               A(IS,L,M)=A(IB,L,M)-A(IB,IP,M)*A(IS,IP,L)
   15    CONTINUE
      ENDIF
C
      RETURN
      END
C
C ****************************************************
C *  SUBROUTINE KEEPIT                               *
C *                                                  *
C *  THIS SUBROUTINE FINDS THE MODEL OF A CANDIDATE  *
C *  REGRESSION                                      *
C ****************************************************
C
      SUBROUTINE KEEPIT(NUMREG,NK,NX,MODELS,Z1,Z3,Z5)
C
      INTEGER Z1,Z3,Z5,NX,NUMREG
      INTEGER NK(Z3),MODELS(Z5,Z1)
C
      DO 10 I=1,NX
         MODELS(NUMREG,I)=NK(I)
   10 CONTINUE
C
      RETURN
      END
C
C ****************************************************
C *   THE FOLLOWING SUBROUTINES, USED IN THIS        *
C *   PROGRAM, ARE IMSL ROUTINES:                    *
C *                                                  *
C *      - BECOVM                                    *
C *           COMPUTES MEANS AND VARIANCE-COVARIANCE *
C *           MATRIX                                 *
```

```
C *      - EIGRF                                        *
C *          COMPUTES EIGENVALUES AND (OPTIONALLY)      *
C *          EIGENVECTORS FOR A REAL GENERAL MATRIX     *
C *          IN FULL STORAGE MODE                       *
C *      - LINV3F                                       *
C *          COMPUTES IN-PLACE INVERSE, EQUATION        *
C *          SOLUTION, AND/OR DETERMINANT EVALUATION    *
C *          IN FULL STORAGE MODE                       *
C *      - MDFD                                         *
C *          COMPUTES F PROBABILITY DISTRIBUTION        *
C *          FUNCTION                                   *
C *      - RLSUBM                                       *
C *          PERFORMS RETRIEVAL OF A SYMMETRIC          *
C *          SUBMATRIX FROM A MATRIX STORED IN          *
C *          SYMMETRIC MODE                             *
C *      - VCVTFS                                       *
C *          PERFORMS STORAGE MODE CONVERSION OF        *
C *          MATRICES (FULL TO SYMMETRIC)               *
C *      - VCVTSF                                       *
C *          PERFORMS STORAGE MODE CONVERSION OF        *
C *          MATRICES (SYMMETRIC TO FULL)               *
C *      - VMULFF                                       *
C *          PERFORMS MATRIX MULTIPLICATION (FULL       *
C *          STORAGE MODE)                              *
C *      - ZREAL2                                       *
C *          COMPUTES THE REAL ZEROS OF A REAL          *
C *          FUNCTION - TO BE USED WHEN INITIAL         *
C *          GUESSES ARE GOOD                           *
C ******************************************************
C
C ***************************************************************
C *      THE FOLLOWING FUNCTIONS ARE USED TO COMPUTE THE        *
C *                     SELECTION CRITERION                     *
C ***************************************************************
C
C ******************************************
C *  FUNCTION C1                           *
C ******************************************
C
      REAL FUNCTION C1(K,IQ,IP)
      PROD=1.
      DO 10 I=1,IP
         ITOP=(K-IQ-I)
         IBOT=(K-IQ-1)*K
         TERM=FLOAT(ITOP)/FLOAT(IBOT)
         PROD=PROD*TERM
   10 CONTINUE
      C1=PROD
      RETURN
      END
C
```

```
C *****************************************
C *  FUNCTION C2                          *
C *****************************************
C
      REAL FUNCTION C2(K,IQ,IP)
      SUM=0.
      P1=1.
      P2=1.
      DO 10 J=0,IP
         ILEFT=JCOMB(IP,J)
         IF (J.NE.0) THEN
            P1=P1*(IQ+2*(J-1))
            P2=P2*(K-IQ-(2*J))
            RNEXT=P1/P2
         ELSE
            RNEXT=1.
         ENDIF
         TERM=FLOAT(ILEFT)*RNEXT
         SUM=SUM+TERM
   10 CONTINUE
      C2=SUM
      RETURN
      END
C
C *****************************************
C *  FUNCTION C3                          *
C *****************************************
C
      REAL FUNCTION C3(K,IQ,IP)
      C3=C1(K,IQ,IP)*C2(K,IQ,IP)
      RETURN
      END
C
C *****************************************
C *  FUNCTION C4                          *
C *****************************************
C
      REAL FUNCTION C4(K,IQ,IP)
      PROD=1.
      DO 10 I=1,IP
         TOP=FLOAT(K-IQ-I)
         BOT=FLOAT(K-IQ-1)
         PROD=PROD*(TOP/BOT)
   10 CONTINUE
      C4=PROD
      RETURN
      END
C
C *****************************************
C *  FUNCTION C5                          *
C *****************************************
C
```

```
      REAL FUNCTION C5(KK,IQ,IP)
      PROD=1.
      DO 10 I=1,IP
         PROD=PROD*(FLOAT(KK-IQ-1)/FLOAT(KK-IQ-I))
   10 CONTINUE
      C5=PROD
      RETURN
      END
C
C ******************************************
C *  FUNCTION CFRONT                       *
C ******************************************
C
      REAL FUNCTION CFRONT(K,IQ,IP)
      TOP=FLOAT(K-IQ-1)
      BOT=FLOAT(K-IQ-IP)
      CFRONT=(TOP/BOT)**IP
      RETURN
      END
C
C ******************************************
C *  FUNCTION F                            *
C ******************************************
C
      REAL FUNCTION F(Z)
      COMMON /BLK1/ SIG,KK,IQQ,IP
      N1=IP
      N2=KK-IQQ-IP
      CALL MDFD(Z,N1,N2,P,IER)
      F=SIG-P
      RETURN
      END
C
C ******************************************
C *  FUNCTION JCOMB                        *
C ******************************************
C
      INTEGER FUNCTION JCOMB(N,M)
      ITOP=NFACT(N)
      IBOT=NFACT(N-M)*NFACT(M)
      JCOMB=ITOP/IBOT
      RETURN
      END
C
C ******************************************
C *  FUNCTION NFACT                        *
C ******************************************
C
      INTEGER FUNCTION NFACT(M)
      IF (M.EQ.0) THEN
         NFACT=1
         RETURN
```

```
      ENDIF
      IP=M
      ILOOP=M-1
      DO 10 I=ILOOP,2,-1
         IP=IP*I
10    CONTINUE
      NFACT=IP
      RETURN
      END
```

APPENDIX B: Documentation for Selection Program

USER'S GUIDE

FOR

VARIABLE SUBSET SELECTION PROGRAM

(VSSP)

Prepared By

Captain James A. Gigliotti

GOR-90M

16 March 1990

TABLE OF CONTENTS

I. INTRODUCTION

Background And Purpose

When dealing with computer simulations it is typically desirable to have a general understanding of how the simulation inputs will affect the final results. It is also desirable to be able to accurately estimate the expected simulation response. Furthermore, if the estimation of the response can be achieved with a subset of the simulation inputs (variables), a variance reduction on the estimator of the mean can also be realized. One way of achieving these goals is through the identification of a good subset of control variates. Control variates, also known as control variables, are variables which have a significant covariance with the response(s) of interest.

The development of a quick and easy method for identifying the subset of significant control variates in a simulation model would greatly decrease the time and effort required to gain insights into the simulation. Identifying the significant control variates for a simulation model can also enhance the process of preparing and implementing an experimental design. It would eliminate the guesswork in determining which variables to concentrate on in a subsequent experimental design. This could also save computer time by identifying a subset of the available control variates to work with, since the standard experimental design requires 2^k simulation runs to acquire data, where k is the number of variables being tested.

The corresponding purpose of the Variable Subset Selection Program (VSSP) is to provide a means to identify the significant control variables, of a simulation, by evaluating the simulation output.

Program Description

The Variable Subset Selection Program (VSSP) identifies the significant control variables using the 'Best Controls' (i.e. BC_p) criterion developed by Bauer and Wilson (1990). Initially, the best or near-best subset of variables, depending on the evaluation procedure desired, is selected for each subset size from 1 to NX (the total number of control variables in the full set). The initial selection among subsets of the same size is based on the RSS (Residual Sums of Squares) of each subset.

When these subsets are selected, the corresponding BC_p criterion value is calculated and the subset with the best criterion value is selected. The major advantage of using the BC_p criterion is that it takes the number of variables in the subset into account in determining the criterion value. Unlike other more common selection criterions, the BC_p criterion does not automatically select the subset with the most variables.

Upon selection of the 'best' variable subset, the corresponding coverage and volume reduction of the confidence region is determined. The values are summed up over each meta-experiment and averaged at the conclusion of the program rum. These values along with the 'best' criterion value and variable subset for each meta-experiment are written to an output file.

As mentioned earlier, the resulting variable subset may be the best or near-best one possible, depending on the selection procedure desired. The user may choose from two selection procedures: 1) Enumerated Subsets, or 2) Stepwise (Forward Selection). The enumerated subsets

procedure evaluates all possible combinations of variables, for each subset size. This ensures that the best subset will be selected, based on the BC_p criterion.

The stepwise (forward selection) procedure does not evaluate all possible subsets, except for the one-variable subset case. Initially, it evaluates all one-variable subsets then selects the best one. After selecting the best single variable the procedure then evaluates only those two-variable subsets containing that one variable; selecting the best of these two-variable subsets. Then the procedure only evaluates the three-variable subsets containing the variables of the best two-variable subset; selecting the best out of those. This process continues, building on the last subset selected, until all variables are in the model. The disadvantage of this procedure is it ignores all subsets which do not include the previously selected variables; so better variable subsets may be missed. However, the advantage of this procedure is that an efficient implementation will select the near-best variable subset in much less time than the enumerated subsets procedure.

In addition, the VSSP provides the user with several more options. The user may specify the following:

- Whether the covariance matrix of control variables should be estimated, based on the data, or provide the covariance matrix, if known.
- Whether the program should provide the single best variable subset or the "M" best X-variable subsets (X = 1,2,...,NX) for each meta-experiment.
- Whether or not program input will be provided by datafile or

input manually.

Brief Overview Of Contents

Following sections of this manual are designed to help the user understand the operation and constructs used in the VSSP. Section II provides information on how to run the VSSP and the options available. Section III lists all the variables used in the program and explains the purpose of each. Section IV provides the parameters associated with each program variable (i.e. variable type, precision (single or double), whether or not it is an array and corresponding size) and any applicable comments. Section V provides a listing of all the subroutines used in the program and a brief description of each. And, Section VI provides a listing of all the functions used in the program and a brief description of each. In addition, several appendixes are provided to supplement the information contained in the various manual sections.

II. HOW TO USE THE PROGRAM AND INTERPRET THE OUTPUT

Program Operation And Required Data

When running the Variable Subset Selection Program (VSSP) the user will be prompted for the following data and information. The minimum required data is a file containing simulation output corresponding to the control variables and responses to be evaluated by the program. The data must be arranged with the variable output values before the response values. Other data which the user may input to the program, if known, is the covariance matrix between the control variables.

The other information, required by the program to operate, is identified in the following list:

- Whether program data and information will be input by datafile or manually. If the datafile option is chosen, the user will be prompted for the datafile name so program data and information can be read in by the program. Regardless, the following information is still required.
- Number of control variables.
- Number of responses.
- Number of best regressions to keep. Input a one (1) if only the best variable subset is desired.
- Number of data replications per meta-experiment.
- Number of meta-experiments.
- Whether or not the covariance matrix of controls is known or is to be estimated from the the data.
- Which evaluation procedure you desire, enumerated subsets or stepwise.

- Level of significance to use in deriving F-values.
- Known or estimated means of the control variables.
- Sample means of the control variables.
- Estimated means of the responses.
- Sample means of the responses.
- A title to write to the program summary file. For this and any other input where character data is necessary, use an underscore '_' instead of spaces. Also, the input must begin with a character, not a number.
- A name for each control variable.
- A name for each response.
- A name for the program summary datafile.

If the covariance matrix of controls is known and is to be provided, the program will ask if this information will be entered by datafile or manually. If the datafile option is selected, then the user will be prompted to provide the name of the datafile where this information is contained. When the covariance matrix is to be provided and the program input is by datafile then the covariance data may be provided by a separate datafile or contained in the initial datafile (See Appendix B for further detail on datafile format).

Finally, there are two items to remember in executing the program or it may not work properly. First, the number of replications per meta-experiment must be greater than the number of control variables plus two (i.e. NX+2). And second, the datafile containing the control and response data must contain, as a minimum, a number of data sets equal to the total replications (i.e. 'Number of meta-experiments' times

'Number of replications per meta-experiment).

Interpreting The Output

After the program has completed evaluation of the data, all results are output to a summary file. The summary file will provide the following information (starting at top-left of file and moving to the right):

- Evaluation title, input by user during program data input phase.
- Number of meta-experiments performed.
- Number of data replications per meta-experiment.
- Total number of data replications evaluated by the program. This is equal to 'Number of meta-experiments' times 'Number of replications per meta-experiment'.
- Summary of response data including response number, name designation of response, response mean over total number of replications, and estimated steady state mean. This information is provided by user.
- Summary of control variable data. Covers same information as for responses. This information is provided by the user.
- A statement whether the covariance matrix of controls was estimated or known.
- The meta-experiment number, best BC_p criterion value found for the meta-experiment, and the corresponding 'best' variable subset.
- Two sets of coverage and volume reduction data averaged over all the meta-experiments. The first set is based on the steady state

means and the second set is based on the sample means.

The following should be noted in regards to the coverage and volume reduction summary data. First, if the steady state and sample means input to the program are the same, there will be no difference in the values of either set. Also, each set of coverage and volume reduction will contain two values for coverage. The primary value of interest is the controlled coverage. This corresponds to data coverage achieved by using the selected variable subsets.

III. VARIABLE DICTIONARY

A = Storage for full covariance matrix and subsequent pivots.

ANSWER = Character variable used to input answers to initial data input questions.

B = Dummy array for IMSL Subroutine LINV3F.

BETA = Equivalent to BUFF2 * SUBV = $\beta = S_{yc} * S_{cc}^{-1}$.

BIG = Equivalent to [(NUMREPS-1)/(NUMREPS-NX-2)]**NY; partial value used in computing TWO.

BOT = Denominater portion of a value used to compute C4 and CFRONT, in their respective functions.

BOUND = Feasibility bound when using known covariance matrix of controls. Used in Subroutine COVER.

BUFF = Buffer used in book keeping of REGR.

BUFF1 = Buffer for S (Full Covariance Matrix), [See Note 1].

BUFF2 = Buffer for S_{yc}, [See Note 1].

BUFF3 = Buffer for S_{cy}, [See Note 1].

BUFF4 = Buffer for S_{yy}, [See Note 1].

BUFF5 = Buffer of SYDOTC.

BUFF6 = Buffer of S_{yy}, [See Note 1].

BUFF9 = Buffer of Sigma Tilde Hat.

C1 = Equivalent to (NUMREPS-2)/(NUMREPS-IIN-1); partial value of residual covariance. Used in Subroutine COVER. Also used in Subroutine COVKNOW to find generalized variance of Sigma Tilde Hat.

C2 = Equivalent to (MV+1)/(NUMREPS(NUMREPS-1)); partial value of generalized variance of Sigma Tilde Hat. Used in Subroutine COVKNOW. Also used in Subroutine COVER in computing the right hand side of confidence region when controls are used.

C3 = Equivalent to (NUMREPS-IIN-1)*(NUMREPS-NY); partial value used in computing volume reduction in Subroutine COVER.

C4 = Equivalent to (NUMREPS-IIN-NY)*(NUMREPS-1); partial value used in computing volume reduction in Subroutine COVER.

C5 = Equivalent to (NY(NUMREPS-1))/(NUMREPS(NUMREPS-NY)); partial value used in computing right hand side for the confidence region where no controls are used. Used in Subroutine COVER.

CANCORR = Canonical Correlations for variable subset models when using a known covariance matrix of controls. Used in Subroutine COVER.

CBAR = Sample mean vector for controls.

CBOT = Denominator of equation used in calculating bound when the covariance matrix of controls is known. Used in MAIN program.

CC1 = Partial value, used to compute actual volume of the controlled and uncontrolled ellipsoids. Used in Subroutine COVER.

CC2 = Partial value, used to compute actual volume of the controlled and uncontrolled ellipsoids. Used in Subroutine COVER.

CC3 = Partial value, used to compute actual volume of the controlled and uncontrolled ellipsoids. Used in Subroutine COVER.

CC4 = Partial value, used to compute actual volume of the controlled and uncontrolled ellipsoids. Used in Subroutine COVER.

CDET = Determinant used in computing volume reduction and actual volume of ellipsoid for case where controls are used and steady state is assumed. Used in Subroutine COVER.

CDEV1 = Equivalent to (CBAR - VECMUC)' = $(\bar{C} - \bar{u}_c)'$.

CDEV2 = Equivalent to (CBAR - VECMUC) = $(\bar{C} - \bar{u}_c)$.

CEIGS = Complex variable counterpart of EIGS.

CONST = Equivalent to (NUMREPS-1)/(NUMREPS-MV-1); partial value used in calculating determinants (DET) in m 'best' regressions mode. Used in MAIN program.

CONST1 = Equivalent to (NUMREPS-2)/(NUMREPS(NUMREPS-1)); partial value used in computing the estimator Sigma Tilde Hat. Used in Subroutine COVER.

CONST2 = Equivalent to (IIN+1)/(NUMREPS(NUMREPS-1)); partial value used in computing the estimator Sigma Tilde Hat. Used in Subroutine COVER.

CONTROL = Character vector which contains names of controls used in the evaluation.

COVCV = Array containing covariance matrix of controls.

COVFILE = Name of datafile containing covariance matrix, if it is known.

COVERAG = Array containing estimated confidence volume coverage.
COVERAG(1): Controlled coverage on steady state means,
COVERAG(2): Uncontrolled coverage on steady state means,
COVERAG(3): Controlled coverage on sample mean, and
COVERAG(4): Uncontrolled coverage on sample mean.

CTOP = Numerator counterpart of CBOT.

D1 = Output of IMSL Subroutine LINV3F. D1 is one of two components of the determininant of the matrix input into LINV3F.

D2 = Output of IMSL Subroutine LINV3F. D2 is one of two components of the determininant of the matrix input into LINV3F.

DET = Determinant of an applicable matrix. Equivalent to D1*2**D2.

DEV = Equivalent to BETA*BUFF3 = $\beta(\bar{C} - u_c)$.

DIFF = Equivalent to VC - VU.

DUM = Dummy array for use in IMSL subroutine calls, when output of that type is not required. Used in MAIN program and Subroutine COVKNOW.

DUMMY = Dummy matrix for use in IMSL subroutine calls, when output of that type is not required. Used in Subroutine COVER.

EHAT = Matrix containing values for estimator Sigma Tilde Hat.

EIGS = Vector variable used in Subroutine Cover to contain eigenvalues derived from IMSL subroutine.

EPS = Covergence criterion used as input to IMSL Subroutine ZREAL2.

EXPL = Equivalent to BETA*BUFF3 = $\beta * S_{cy}$.

F = Used in computing right hand side of confidence region, in Subroutine COVER.

F2 = Used in computing right hand side of volume reduction, in Subroutine COVER.

FF = Contains F^p table.

FP = Equivalent to ROOT(1)**IP. Used in Subroutine FTABL to hold F-value until it is added to array FF.

FULL = Full storage mode version of VCV.

FULCOVC = Full covariance matrix of all controls and responses.

GAMM = Gamma Hat matrix, used in Subroutine COVER.

GAMMA = Used to compute actual volume of controlled ellipsoid, in Subroutine COVER.

HPH = Equivalent to T1*CDEV2 = $(\tilde{\bar{C}} - \tilde{u}_C)'*S_C^{-1}*(\tilde{\bar{C}} - \tilde{u}_C)$.

I = Counting variable used in DO loops. Also used to count number of primary variable inputs which exceed program parameters, if any.

I1 = DO loop counting variable, used in Stepwise section of MAIN program.

I2 = DO loop counting variable, used in Stepwise section of MAIN program.

IAT = Tracks which regression model is currently being evaluated.

IB = Index of source block. Used in Subroutine GAUSS.

IB2 = Provides same function as IB. Used in Stepwise procedure of MAIN program.

IBOT = Denominator for values computed in Functions C1 and JCOMB.

IBUFF = Array containing control variables in 'best' regression model selected. Identifies variables by number, not model coefficients.

IC = Counting variable used in DO loops.

ICOUNT = Acts as reference value, in Subroutine COVER, in computing canonical correlations when covariance matrix of controls is known.

ICOVER = Indicator array of coverage for a particular model; 0=No, 1=Yes.

ICOVER(1) = 0,1 (Controls present, steady state assumed)
ICOVER(2) = 0,1 (No controls, steady state assumed)
ICOVER(3) = 0,1 (Controls present, $\bar{Y}$(1000))
ICOVER(4) = 0,1 (No controls, $\bar{Y}$(1000))

ICTOT = Keeps coverage total as each meta-experiment is performed. Used in MAIN program in computing average coverage over all meta-experiments; COVERAG(I)=ICTOT(I)/META.

IER = Error condition, output from IMSL subroutines if an error condition is encountered.

IH = Inclusion array, of both controls and responses, for input to IMSL Subroutine RLSUBM. Used in MAIN program and Subroutine COVER.

IH2 = Inclusion array, of controls only, for submatrix of selected model used as input to IMSL Subroutine RLSUBM. Used in Subroutine COVER.

II = Counting variable used in DO loops.

IIN = Number of variables in current model.

IKNOW = Flag designating whether the covariance matrix is estimated (IKNOW=0), or known (IKNOW=1).

ILEFT = Number of pivots left to process in computing Function C2; ILEFT = NFACT(IP)/(NFACT(IP-J)*NFACT(J)).

ILOOP = Equivalent to M-1 in Function NFACT. Used as starting point for loop which computes the factorial of M.

IND1 = Array for keeping 'best' evaluated subset of size J=1...NX. Used in Stepwise procedure of MAIN program.

IND2 = Array for maintaining latest subset created for evaluation within the Stepwise procedure section of MAIN program.

INDEX = Index variable for SCBAR and SVECMU when computing subvector of the control means in Subroutine COVER.

INFILE = Character variable used in initial data input routine. INFILE takes name of input file if datafile option is chosen.

IORDER = Order (number of rows) of input or output matrix. Used in several IMSL Subroutines.

IP = Index of the pivot row and column, used in Subroutine GAUSS.

IQ = Counting variable used in DO loops.

IQQ = Counting variable used in DO loops.

IS = Index of the storage block, used in Subroutine GAUSS.

IS2 = Provides same function as IS. Used in Stepwise procedure of MAIN program.

ITMAX = Input to IMSL Subroutine ZREAL2, defines the maximum number of iterations to use in finding a root.

ITOP = Numerator for values computed in Functions C1 and JCOMB.

IVAR = Integer value associated with the individual control variables (i.e. 1=X1, 2=X2, etc.).

IWRITE = Flag designating whether the meta experiment mode (IWRITE=0) or best 'm' regressions mode (IWRITE=1) is to be used.

IX = Equivalent to Z6, used as input to IMSL Subroutine BECOVM.

IZ = Counting variable used in DO loops. Used in initializing arrays and matrices in meta loop of MAIN program.

J = Counting variable used in DO loops.

JJ = Counting variable used in DO loops.

JZ = Counting variable used in DO loops. Used in initializing arrays and matrices in meta loop of MAIN program.

K = See NX.

KEEPERS = Number of 'best' regressions to keep (See M).

KK = See NUMREPS.

KNX = Equivalent to 2**NX.

KP = Equivalent to k+1, where k is number of control variates. Used in Subroutine GAUSS.

KZ = Counting variable used in DO loops. Used in initializing matrices in meta loop of MAIN program.

L = Counting variable used in DO loops.

LAST = Variable used as input to IMSL Subroutine ZREAL2, contains initial guess of root for defined function.

LB = Equivalent to IP+1, used in Subroutine GAUSS.

M = Counting variable used in DO loops.

M1 = Variable used as input to IMSL Subroutine RLSUBM, contains order of symmetric matrix stored in symmetric mode.

M2 = Order of symmetric matrix SUBV. Obtained as output of IMSL Subroutine RLSUBM, and used as input to IMSL Subroutine VCVTSF. Used in Subroutine COVER.

MAX = Maximum number of storage matrices in Array A, for storage of matrices created by calls to Subroutine GAUSS. Used in Stepwise section of MAIN program.

META = Number of meta experiments. This is required input data.

METHOD = Flag designating whether enumerated subsets (METHOD=0) or stepwise [forward selection] (METHOD=1) method is to be used in the evaluation. This is required input data.

MM = Used as DO loop counting variable for META loop.

MODELS = Saves all the 2^{NX} enumerated subset models.

MV = Counts number of control variables contained in a model as defined by NK. Used in MAIN program and Subroutine COVKNOW.

N = Counting variable used in DO loops.

NBR = Vector of inputs to IMSL Subroutine BECOVM.

NK = An identification array, NK is a binary counter with a list of zeros and ones which indicate the presence or absence of the independent variables (i.e. controls). Also note that indexing of the independent variables is reversed.

NROOT = Input to IMSL Subroutine ZREAL2, defines number of roots to be found.

NSIG = Convergence criterion input to IMSL Subroutine ZREAL2. A root is accepted if two successive approximations to a given root agree in the first NSIG digits.

NUMREG = Tracks the number of 'best' regressions, when that mode is used.

NUMREPS = Number of replications per meta experiment. This is required input data.

NVAR = Total number of variables (i.e. NVAR = NX + NY).

NX = Number of candidate control variates. This is required data input.

NY = Number of responses. This is required input data.

OBS = Equivalent to T2*YMD2.

OBS2 = Output of IMSL Subroutine VMULFF, contains product of the first two matrices provided as input to the subroutine. Also used in determining coverage when no controls are used and steady state is assumed. Used in Subroutine COVER.

OUTFILE = Character variable used input name of an input datafile created during manual data input, if desired. Also used to input name for file to contain program output.

P = Output probability, of IMSL Subroutine MDFD, that a random variable following the F-Distribution with degrees of freedom N1 and N2 will be less than or equal to input Z. Used in Function F. This value supports calculation of F-table in Subroutine FTABL.

P1 = Product 1, numerator used in computing RNEXT, in Function C2. P1 = [PROD(J=0,IP) (IQ+(2*(J+1)))].

P2 = Product 2, denominator used in computing RNEXT, in Function C2. P2 = [PROD(J=0,IP) (K-IQ-(2*J))].

PI = Parameter in Subroutine Cover which contains value of pi.

POVER2 = Equivalent to NY/2. Used in Subroutine COVER to compute actual volume of controlled and uncontrolled ellipsoid.

PROD = Keeps cumulative product of terms, for Functions C1, C4, and C5.

RDET = Array of determinants of matrices associated with the specific regression models.

REGR = Bookkeeping array for best M regressions to keep. For array of format REGR(i,j,k); A) j = subset size, B) k=1, stores generalized matrix; and k=2, stores pointer to model.

REIGS = Equivalent to Sqrt(EIGS(ICOUNT)), used in computing the canonical correlations and feasibility bound when the covariance matrix of controls is known. Used in Subroutine COVER.

RESPONS = Character vector used to contain names of responses used in the evaluation.

RHS = Right Hand Side of the confidence region, where controls are used, per Rao (1967). Used in Subroutine COVER.

RHS2 = Right Hand Side of the confidence region, when no controls are used and steady state assumed. Used in Subroutine COVER.

RMIN = Holds minimum Residual Sums of Squares (RSS) values. RSS values used in determining the m 'best' regressions, when that mode is selected.

RNEXT = Equivalent to P1/P2 in Function C2. Used in computing TERM.

ROOT = Used as input/output to IMSL Subroutine ZREAL2, in Subroutine FTABL. As input, contains initial guess of root. As output, contains computed root.

RSS = Buffer for conditional covariance matrix.

SCBAR = Subvector of CBAR, used to find vector of corrections to control $\bar{Y}$ (variables CDEV1 and CDEV2). Used in Subroutine COVER.

SIG = Level of Significance associated with selection criteria.

SP = Equivalent to RMIN, used in write statement to program output file.

SUBCOVC = Submatrix of full covariance matrix FULCOVC.

SUBV = Submodel covariance matrix in symmetric storage.

SUBVF = Full storage version of $SUBV^{-1}$.

SUM = Keeps sum of TERMs in Function C2.

SUMDEV = Keeps sum of DIFFs for each meta-experiment.
SUMDEV(1): For steady state means.
SUMDEV(2): For sample means.

SUMVU = Keeps sum of VUs for each meta-experiment.
SUMVU(1): For steady state means.
SUMVU(2): For sample means.

SVECMU = Subvector of VECMUC, used to find vector of corrections to control $\bar{Y}$. Used in Subroutine COVER.

SYDOTC = Equivalent to BUFF4-EXP = $C2(S_{yy} - ß*S_{cy})$.

SYMCOVC = Symmetric storage version of matrix FULCOVC.

T1 = Equivalent to CDEV1*SUBVF = $(\tilde{\bar{C}} - \tilde{u}_c)'*S_c^{-1}$.

T2 = Equivalent to YMD1*SYDOTC.

TARGET = Matrix used in computing generalized variance of Sigma Tilde Hat (i.e. responses).

TEMP = Input vector of length NBR(1), used in IMSL Subroutine BECOVM. If NBR(5)=0, then TEMP must contain the temporary means when NBR(4)=1. Otherwise, temp is work storage.

TERM = Generic variable, used in Functions C1 and C2 to save results of divisions and products, respectively.

TERM1 = Partial value used in computing volume reduction where controls are used and steady state assumed. Used in Subroutine COVER.

TERM2 = Partial value used in computing volume reduction where controls are used and steady state assumed. Used in Subroutine COVER.

TERM3 = Partial value used in computing volume reduction where controls are used and steady state assumed. Used in Subroutine COVER.

TIND = Array for keeping current/temporary 'best' variable subset model as all models are created for evaluation in Stepwise section of MAIN program. When all models of size J=1,...,NX have been evaluated, the model maintained in TIND is copied to IND1, as the 'best' subset of size J.

TITLE = Analysis title, input during initial data input routine and written to output file.

TMV = Tracks number of variables in last variable subset evaluated in the Stepwise procedure section of MAIN program. Acts as a flag to trigger save of best subset of a certain size.

TOP = Numerator portion of a value used to compute C4, in Function C4.

TWO = Modified version of matrix determinant, equivalent to 2*BIG*DET. Used as determinant bound and as original values for elements of bookkeeping array REGR.

UCDET = Determinant used in computing the volume reduction where controls are used and steady state assumed, and the actual volume of the uncontrolled ellipsoid. Used in subroutine COVER.

VC = Volume of controlled ellipsoid, used in Subroutine COVER.

VCV = Covariance matrix in symmetric storage of all controls and responses. This is an output of IMSL Subroutine BECOVM.

VECCBAR = Vector of average of inputs of controls.

VECMUC = Vector of theoretical means of the controls.

VECMUY = Vector of steady state means of the responses.

VECYBAR = Vector of sample means of the responses.

VOLRED = VOLRED(1) is volume reduction due to controls; VOLRED(2) is not used.

VR = Array of Variance Reduction values, derived from calulating selection criterion for each regression model.

VU = Volume of uncontrolled ellipsoid, used in Subroutine COVER.

WK = Array used by IMSL Subroutines EIGRF. Provides work area for subroutine to use in performing its function.

WKAREA = Array used by IMSL Subroutine LINV3F. Provides same function as WK.

X = Data matrix for a single meta-experiment.

XFILE = Datafile containing [controls¦response] data.

XM = Output vector, of length NBR(1), of IMSL Subroutine BECOVM. This vector contains the variable means.

YBAR = Sample mean vector for controls.

YBHAT = Equivalent to $\bar{Y} - ß(\bar{C} - u_c) = \tilde{Y}(ß)$.

YMD1 = Equivalent to $(\tilde{Y}(\hat{ß}) - u_y)'$.

YMD2 = Equivalent to $(\tilde{Y}(\hat{ß}) - u_y)'$.

YMD3 = Equivalent to (YBAR - VECMUY)' = $(\bar{Y} - u_y)'$.

YMD4 = Equivalent to (YBAR - VECMUY) = $(\bar{Y} - u_y)$.

Z = Input constant, to IMSL Subroutine MDFD, to which integration is performed. Z must be greater than or equal to zero. Used in Function F.

Z1 = Program parameter defining maximum number of Control Variables (See NX) the program is set to handle. This parameter is also used in Subroutine Cover.

Z2 = Program parameter defining maximum number of Control Responses (See NY) the program is set to handle. This parameter is also used in Subroutine Cover.

Z3 = Program parameter equivalent to Z1+Z2. This parameter is also used in Subroutine Cover.

Z4 = Program parameter defining maximum number of 'best' regressions which may be kept.

Z5 = Program parameter equivalent to 2**Z1. This parameter also used in Subroutine Cover.

Z6 = Program parameter defining maximum number of replications per meta-experiment allowed.

Z7 = Program parameter defining maximum number of meta experiments allowed.

Z8 = Program parameter equivalent to (Z3*(Z3+1))/2. This parameter also used in Subroutine Cover.

NOTES: (1) $S = \left(\begin{array}{c|c} S_{cc} & S_{cy} \\ \hline S_{yc} & S_{yy} \end{array} \right)$

IV. VARIABLE PARAMETERS MATRIX

VARIABLE	Variable Type: INT	REAL	CHAR	COM	LOGIC	ARRAY	Precision: S	D	Size/Comments
A		X				X	X		MAXxZ3xZ3
ANSWER			X						
B		X				X	X		Z3
BETA		X				X	X		Z2xZ1
BIG		X					X		
BOT		X					X		
BOUND		X					X		
BUFF		X				X	X		Z4
BUFF1		X				X	X		Z3xZ3
BUFF2		X				X	X		Z4 (MAIN) Z2xZ1(COVER)
BUFF3		X				X	X		Z1xZ2
BUFF4		X				X	X		Z2xZ2
BUFF5		X				X	X		Z2xZ2
BUFF6		X				X	X		Z2xZ2
BUFF9		X				X	X		Z2xZ2
C1		X					X		
C2		X					X		
C3		X					X		
C4		X					X		
C5		X					X		
CANCORR		X				X	X		Z2xZ2
CBAR		X				X	X		Z1
CBOT		X					X		

VARIABLE PARAMETERS MATRIX
(Continued)

	Variable Type						Precision		
VARIABLE	INT	REAL	CHAR	COM	LOGIC	ARRAY	S	D	Size/Comments
CC1		X					X		
CC2		X					X		
CC3		X					X		
CC4		X					X		
CDET		X					X		
CDEV1		X				X	X		1xZ1
CDEV2		X				X	X		Z1x1
CEIGS				X		X			Z2
CONST		X					X		
CONST1		X					X		
CONST2		X					X		
CONTROL			X			X			C:25 A:Z1
COVCV		X				X	X		Z1xZ1
COVFILE			X						25
COVERAG		X				X	X		4
CTOP		X					X		
D1		X					X		
D2		X					X		
DET		X					X		
DEV		X				X	X		Z2x1
DIFF		X					X		
DUM		X				X	X		Z2
DUMMY		X				X	X		Z2xZ2

VARIABLE PARAMETERS MATRIX
(Continued)

VARIABLE	Variable Type INT	REAL	CHAR	COM	LOGIC	ARRAY	Precision S	D	Size/Comments
EHAT		X				X	X		Z2xZ2
EIGS		X				X	X		2*Z2
EPS		X					X		
EXPL		X				X	X		Z2xZ2
F		X					X		
F2		X					X		
FF		X				X	X		0:Z1
FP		X					X		
FULL		X				X	X		Z3xZ3
FULCOVC		X				X	X		Z1xZ1
GAMM		X				X	X		Z2xZ1
HPH		X				X	X		1x1
I	X								
I1	X								
I2	X								
IAT	X								
IB	X								
IB2	X								
IBOT	X								
IBUFF	X					X			Z1
IC	X								
ICOUNT	X								
ICOVER	X								

VARIABLE PARAMETERS MATRIX
(Continued)

VARIABLE	INT	REAL	CHAR	COM	LOGIC	ARRAY	Precision S	Precision D	Size/Comments
ICTOT	X					X			4
IER	X								
IH	X					X			Z3
IH2	X					X			Z1
II	X								
IIN	X								
IKNOW	X								
ILEFT	X								
ILOOP	X								
IND1	X					X			Z1+1
IND2	X					X			Z1+1
INDEX	X								
INFILE			X						25
IORDER	X								
IP	X								
IQ	X								
IQQ	X								
IS	X								
IS2	X								
ITMAX	X								
ITOP	X								
IVAR	X								
IWRITE	X								

VARIABLE PARAMETERS MATRIX
(Continued)

VARIABLE	Variable Type						Precision		Size/Comments
	INT	REAL	CHAR	COM	LOGIC	ARRAY	S	D	
IX	X								
IZ	X								
J	X								
JJ	X								
JZ	X								
K	X								
KEEPERS	X								
KK	X								
KNX	X								
KP	X								
KZ	X								
L	X								
LAST		X					X		
LB	X								
M	X								
M1	X								
M2	X								
MAX	X								Program Parameter
META	X								
METHOD	X								
MM	X								
MODELS	X					X			Z5xZ1

VARIABLE PARAMETERS MATRIX
(Continued)

VARIABLE	INT	REAL	CHAR	COM	LOGIC	ARRAY	Precision S	Precision D	Size/Comments
MV	X								
N	X								
NBR	X					X			6
NK	X					X			Z3
NROOT	X								
NSIG	X								
NUMREG	X								
NUMREPS	X								
NVAR	X								
NX	X								
NY	X								
OBS		X				X	X		1x1
OBS2		X				X	X		1x1
OUTFILE			X						25
P		X					X		
P1		X					X		
P2		X					X		
PI		X					X		
PROD		X					X		
RDET		X					X		
REGR		X				X	X		Z4xZ1x2
REIGS		X				X	X		Z2
RESPONS			X			X			C:25 A:Z2

VARIABLE PARAMETERS MATRIX
(Continued)

	Variable Type						Precision		
VARIABLE	INT	REAL	CHAR	COM	LOGIC	ARRAY	S	D	Size/Comments
RHS		X					X		
RHS2		X					X		
RMIN		X					X		
RNEXT		X					X		
ROOT		X				X	X		1
RSS		X				X	X		Z2xZ2
SCBAR		X				X	X		Z1
SIG		X					X		
SP		X					X		
SUBCOVC		X				X	X		(Z1(Z1+1))/2
SUBV		X				X	X		Z8
SUBVF		X				X	X		Z3xZ3
SUM		X					X		
SUMDEV		X				X	X		2
SUMVU		X				X	X		2
SVECMU		X				X	X		Z1
SYDOTC		X				X	X		Z2xZ2
SYMCOVC		X				X	X		(Z1(Z1+1))/2
T1		X				X	X		1xZ1
T2		X				X	X		1xZ2
TARGET		X				X	X		Z2xZ2
TEMP		X				X	X		Z3
TERM		X					X		

VARIABLE PARAMETERS MATRIX
(Continued)

VARIABLE	Variable Type: INT	REAL	CHAR	COM	LCGIC	ARRAY	Precision: S	D	Size/Comments
TERM1		X					X		
TERM2		X					X		
TERM3		X					X		
TIND	X					X			Z1+1
TITLE			X						25
TOP		X					X		
TWO		X					X		
UCDET		X					X		
VC		X					X		
VCV		X				X	X		(Z3(Z3+2))/2
VECCBAR		X				X	X		Z1
VECMUC		X				X	X		Z1
VECMUY		X				X	X		Z2
VECYBAR		X				X	X		Z2
VOLRED		X				X	X		2
VR		X				X	X		2
VU		X					X		
WK		X				X	X		Z2
WKAREA		X				X	X		Z2
X		X				X	X		Z6xZ3
XFILE			X						25
XM		X				X	X		Z3
YBAR		X				X	X		Z2

VARIABLE PARAMETERS MATRIX
(Continued)

VARIABLE	Variable Type: INT	REAL	CHAR	COM	LOGIC	ARRAY	Precision: S	D	Size/Comments
YBHAT		X				X	X		Z2
YMD1		X				X	X		1xZ2
YMD2		X				X	X		Z2x1
YMD3		X				X	X		1xZ2
YMD4		X				X	X		Z2x1
Z		X					X		
Z1	X								Program Parameter
Z2	X								Program Parameter
Z3	X								Program Parameter
Z4	X								Program Parameter
Z5	X								Program Parameter
Z6	X								Program Parameter
Z7	X								Program Parameter
Z8	X								Program Parameter

V. SUBROUTINE LISTING AND DESCRIPTION

BECOVM	=	An IMSL subroutine which computes means and variance-covariance matrix.
COVER	=	This subroutine does the coverage and volume reduction calculations for the optimal control subset.
COVKNOW	=	This subroutine returns the generalized variance of sigma tilde hat.
EIGRF	=	An IMSL subroutine which computes eigenvalues and (optionally) eigenvectors of a real general matrix in full storage mode.
FTABL	=	This subroutine computes an F table, to the power P.
GAMMA	=	An implicit FORTRAN function which provided Gamma Function value.
GAUSS	=	This subroutine performs the pivots for variable introduction into regression models. (Furnival and Wilson 1974)
KEEPIT	=	This subroutine finds the model of a candidate regression.
LINV3F	=	An IMSL subroutine which computes in-place inverse, equation solution, and/or determinant evaluation, uses full storage mode.
MDFD	=	An IMSL subroutine which computes the F probability distribution function.
RLSUBM	=	An IMSL subroutine which performs retrieval of a symmetric submatrix from a matrix stored in symmetric mode by RLSTP.
VCVTFS	=	An IMSL subroutine which performs storage mode conversion of matrixes (full to symmetric).
VCVTSF	=	An IMSL subroutine which performs storage mode conversion of matrixes (symmetric to full).
VMULFF	=	An IMSL subroutine which performs matrix multiplication (full storage mode).
ZREAL2	=	An IMSL subroutine which computes the real zeros of a real function - to be used when initial guesses are good.

Note: The following subroutine is not used in program, but is referenced in description of subroutine RLSUBM. In addition, complete IMSL subroutine usage descriptions are provided in Attachment D.

[RLSTP] = An IMSL subroutine which performs regression model selection using a forward stepwise algorithim, with results available after each step.

VI. FUNCTION LISTING AND DESCRIPTION

C1 = Equivalent to [PROD(I=1,IP) ((K-IQ-I)/(K(K-IQ-1))]. Partial value used in computing selection criterion for a model, when the covariance matrix of controls is estimated. Used in Function C3.

C2 = Equivalent to {1 + [SUM(J=1,P) JCOMB(P,J)*(Q(Q+2)...(Q+2(J-1))) / ((K-Q-2)...(K-Q-2J))]}. Partial value used in computing the selection criterion when the covariance matrix is estimated. Used in Function C3.

C3 = Equivalent to C1 * C2. Partial value used in computing selection criterion for a model, when covariance matrix of controls is estimated. This provides a loss factor for the measure of efficiency of control variables as defined by Rubenstein and Marcus (1985). Used in MAIN program.

C4 = Equivalent to [PROD(I=1,IP) ((K-IQ-I)/(K-IQ-1))]. Partial value used in computing selection criterion for a model, when covariance matrix of controls is known. Used in MAIN program.

C5 = Equivalent to [PROD(I=1,IP) ((KK-IQ-I)/(KK-IQ-1))]. Partial value used in computing selection criterion for a model, regardless of whether covariance matrix is known or estimated. Used in MAIN program.

CFRONT = Equivalent to I*[(K-IQ-1)/(K-IQ-IP)]. This value is used in calculating the 100(1-a)% confidence ellipsoid about the responses as defined by Rao (1967). Used in MAIN program.

F = A single-arguement real function subprogram used by IMSL Subroutine ZREAL2. F defines the function for which the roots are to be found.

JCOMB = Equivalent to [NFACT(N)/(NFACT(N-M)*NFACT(M))], which is the number of possible combinations of N items taken M at a time.

NFACT = Computes factorial of X (i.e. X! = X*(X-1)*(X-2)*...*1).

ATTACHMENT A: VARIABLE SUBSET SELECTION PROGRAM CODE (FORTRAN)

This program listing is already provided as Appendix A to the thesis.

ATTACHMENT B: FORMAT FOR DATA INPUT FILES

In the event that program data will be input through the use of a datafile, the following format is required. It should be noted that the data can be placed as shown or on seperate lines for each value see examples 1 and 2, respectively), the order remains as shown. In addition, a combination of these two formats may be used, if desired.

```
NX NY KEEPERS NUMREPS META
IKNOW IWRITE METHOD
SIG
VECMUC  (NX)
VECCBAR (NX)
VECMUY  (NY)
VECYBAR (NY)
TITLE
CONTROL (NX)
RESPONS (NY)
<COVFILE; IF IKNOW=1>
<COVCV (NXxNX); IF COVFILE=INFILE AND IKNOW=1>
XFILE
```

NOTES:

[1] VARIABLE NAME = SPECIFIC DATA TO PLACE AT THAT POINT.

[2] (X1) = X1 ROWS, OR TOTAL OF X1 ELEMENTS, EXPECTED.

[3] (X1xX2) = X1 ROWS BY X2 COLUMNS, OR TOTAL OF X1xX2 ELEMENTS, EXPECTED.

[4] <X;Y> = COMMENTS ON OPTIONAL FILE DATA. X IS REQUIRED DATA IF CONDITION Y IS MET, OTHERWISE DO NOT INSERT THIS DATA INTO THE DATAFILE.

[5] DO NOT PLACE COMMAS BETWEEN DATA VALUES.

DATAFILE FORMAT EXAMPLES:

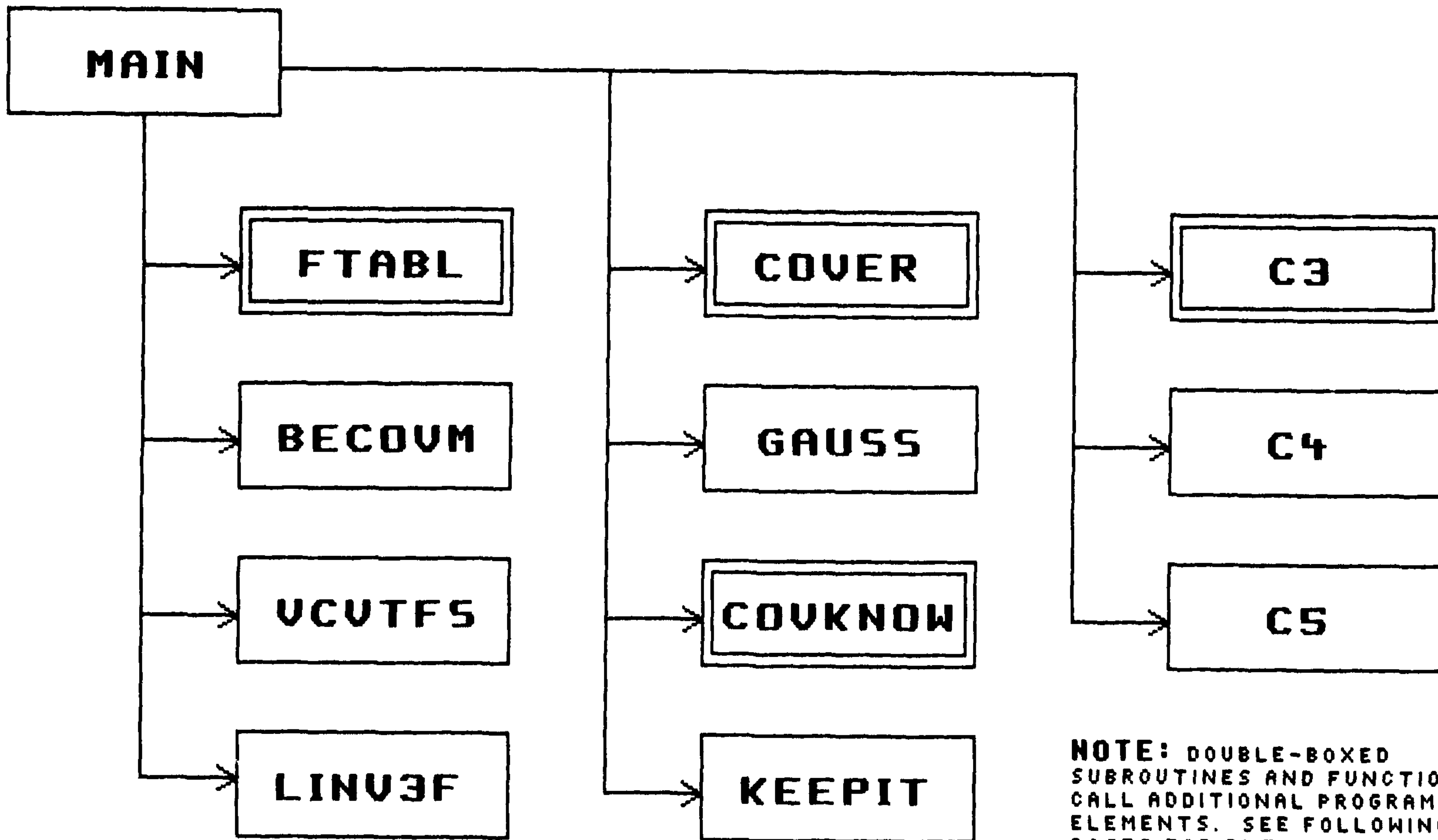
Example 1:

```
5  2  1  10  100
0  0  0
0.90
0  0  0  0  0
0  0  0  0  0
0  0
0  0
EXAMPLE_1
C1  C2  C3  C4  C5
R1  R2
SUMMARY_EX#1
```

Example 2:

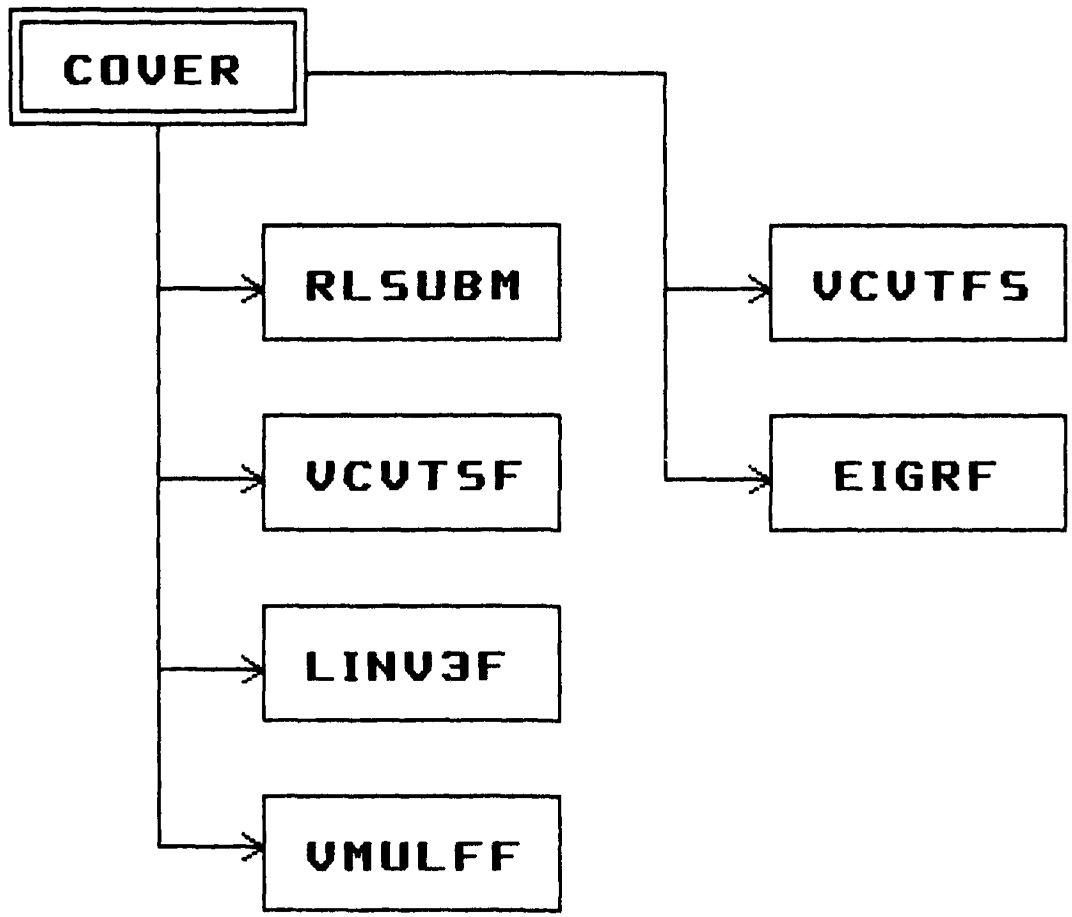
```
5
2
1
10
100
0
0
0
0.90
0
0
0
0
0
0
0
0
0
0
0
0
0
0
EXAMPLE_2
C1
C2
C3
C4
C5
R1
R2
SUMMARY_EX#2
```

ATTACHMENT C: PROGRAM FLOW DIAGRAMS

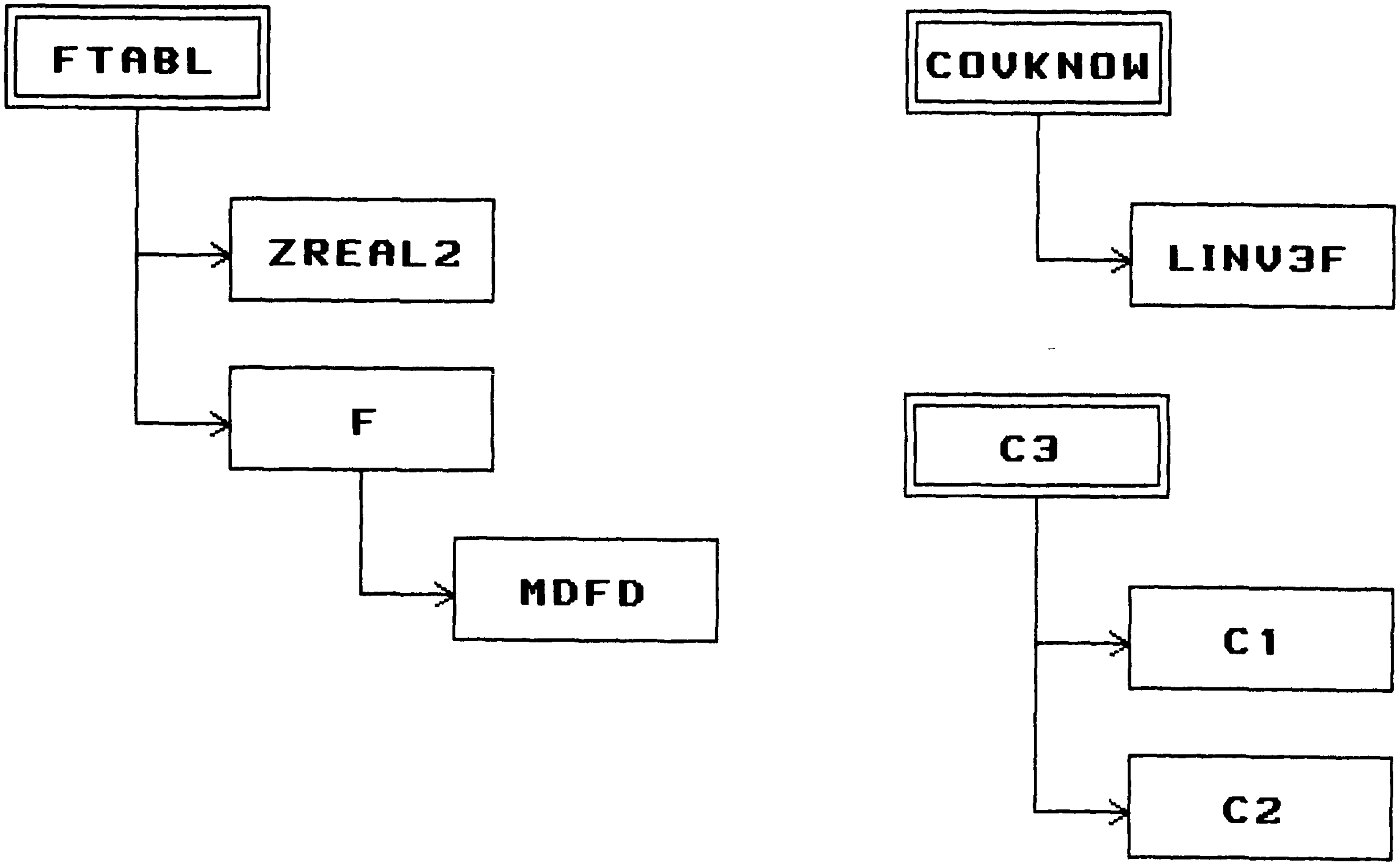


NOTE: DOUBLE-BOXED SUBROUTINES AND FUNCTIONS CALL ADDITIONAL PROGRAM ELEMENTS. SEE FOLLOWING PAGES FOR DETAILS.

Subroutines And Functions Called By Program MAIN



Program Elements Called By Subroutine COVER



Program Elements Called By COVKNOW, FTABL, and C3

ATTACHMENT D: IMSL SUBROUTINE USAGE DESCRIPTIONS

```
IMSL ROUTINE NAME  - BECOVM

PURPOSE            - Means and Variance-Covariance matrix

USAGE              - CALL BECOVM(X,IX,NBR,TEMP,XM,VCV,IER)

ARGUEMENTS  X      - ON INPUT: X is an NBR(3) by NBR(1) submatrix of the
                        matrix (call it XX) of data for which means,
                        variances and covariances, or corrected sums of
                        squares and cross products are desired.  The
                        last submatrix in XX may have fewer than NBR(3)
                        rows.
                     ON OUTPUT: The rows of X have been adjusted by the
                        temporary means.

            IX     - Input, row dimension of X exactly as specified in
                        the dimension statement in the calling program.

            NBR    - Input vector of length 6.  NBR(I) contains, when
                        I=1, Number of variables.
                        I=2, Number of observations per variable in XX.
                        I=3, Number of observations per variable in each
                           submatrix X, not including the last submatrix
                           where the number may be less than or equal to
                           NBR(3).  However, NBR(3) should be the same
                           for all calls.
                        I=4, The number of the submatrix stored in X.
                        I=5, The temporary mean indicator.  If NBR(5)=0,
                           the user supplies temporary means in TEMP.
                           Otherwise, the first row of XX (or first row
                           of X when NBR(4)=1) is utilized.
                        I=6, The VCV option.  If NBR(6)=0, VCV contains
                           the Variance-Covariance matrix.  Otherwise,
                           VCV contains the corrected cross sums of
                           squares and cross-products matrix.

            TEMP   - Input vector of length NBR(1).  If NBR(5)=0, TEMP
                        must contain the temporary means when NBR(4)=1.
                        Otherwise, TEMP is work storage.

            XM     - Output vector of length NBR(1) containing the
                        variable means.

            VCV    - Output NBR(1) by NBR(1) matrix stored in symmetric
                        storage mode requiring (NBR(1)*(NBR(1)+1))/2
                        storage locations.  VCV contains the Variance-
                        Covariance matrix or the corrected sums of
                        squares and cross-products matrix, as controlled
                        by the VCV option, NBR(6).
```

```
IER      - Error parameter. (Output)
           Terminal Error:
              IER=129, Indicates that NBR(4) is less than 1 or
                 NBR(3)*(NBR(4)-1) exceeds NBR(2).
              IER=130, Indicates that NBR(1) is less than 1 or
                 NBR(2) is less than 2 or NBR(3) exceeds
                 NBR(2).

PRECISION/HARDWARE - Single and Double/H32
                   - Single/H36,H48,H60

REQD IMSL ROUTINES - UERTST, UGETIO

NOTATION           - Information on special notation and conventions is
                     available in the manual or through IMSL routine
                     UHELP.
```

IMSL ROUTINE NAME - EIGRF

PURPOSE - Eigenvalues and (optionally) eigenvectors of a real general matrix in full storage mode.

USAGE - CALL EIGRF(A,N,IA,IJOB,W,Z,IZ,WK,IER)

ARGUEMENTS

A - The input real general matrix of order N whose eigenvalues and eigenvectors are to b ecomputed. Input A is destroyed if IJOB is equal to 0 or 1.

N - The input order of the matrix A.

IA - The input row dimension of matrix A exactly as specified in the dimension statement in the calling program.

IJOB - The input option parameter. When
IJOB=0, Compute eigenvalues only.
IJOB=1, Compute eigenvalues and eigenvectors.
IJOB=2, Compute eigenvalues, eigenvectors, and performance index.
IJOB=3, Compute performance index only. If the performance index is computed, it is returned in WK(1). The routines have performed (Well, Satisfactorily, Poorly) if WK(1) is (Less than 1, Between 1 and 100, greater than 100).

W - The output complex vector of length N, containing the eigenvalues of A.

Z - The output N by N complex matrix containing the eigenvectors of A. The eigenvector in column J of Z corresponds to the eigenvalue W(J). If IJOB=0, Z is not used.

IZ - The input row dimension of matrix Z exactly as specified in the dimension statement in the calling program. IZ must be greater than or equal to N if IJOB is not equal to zero.

WK - Work area, the length of WK depends on the value of IJOB, when
IJOB=0, The length of WK is at least N.
IJOB=1, The length of WK is at least 2N.
IJOB=2, The length of WK is at least (2+N)N.
IJOB=3, The length of Wk is at least 1.

```
IER      - Error parameter. (Output)
           Terminal Error:
              IER=128+J, Indicates that EQRH3F failed to
                 converge on eigenvalue J.  Eigenvalues J+1,
                 J+2, ..., N have been computed correctly.
                 Eigenvalues 1,...,J are set to zero.
              IER=1 or 2, Eigenvectors are set to zero.  The
                 performance index is set to 1000.
           Warning Error (with fix)
              IER=66, Indicates IJOB is less than 0 or IJOB is
                 greater than 3.  IJOB set to 1.
              IER=67, Indicates IJOB is not equal to zero, and
                 IZ is less than the order of matrix A.  IJOB
                 is set to zero.

PRECISION/HARDWARE - Single and Double/H32
                   - Single/H36,H48,H60

REQD IMSL ROUTINES - EBALAF, EBBCKF, EHBCKF, EHESSF, EQRH3F, UERTST,
                     UGETIO

NOTATION           - Information on special notation and conventions is
                     available in the manual introduction or through
                     IMSL routine UHELP.
```

```
IMSL ROUTINE NAME  - LINV3F

PURPOSE            - In place inverse, equation solution, and/or
                     determinant evaluation - full storage mode.

USAGE              - CALL LINV3F(A,B,IJOB,N,IA,D1,D2,WKAREA,IER)

ARGUEMENTS   A     - Input/output matrix of dimension N by N. See
                        parameter IJOB.

             B     - Input/output vector of length N when IJOB=2 or 3.
                        Otherwise, B is not used.
                     On input: B contains the right hand side of the
                        equation AX = B.
                     On output: The solution X replaces B.

             IJOB  - Input option parameter. IJOB=I implies:
                        I=1, Invert matrix A.  A is replaced by its
                           inverse.
                        I=2, Solve the equation AX=B.  A is replaced by
                           the LU decomposition of a rowwise permutation
                           of A, where U is upper triangular and L is
                           lower triangular with unit diagonal.  The
                           unit diagonal of L is not stored.
                        I=3, Solve AX=B and invert matrix A.
                        I=4, Compute the determinant of A.  A is
                           replaced by the LU decomposition of a rowwise
                           permutation of A.

             N     - Order of A. (Input)

             IA    - Row dimension of matrix A exactly as specified in
                        the dimension statement in the calling program.
                        (Input)

             D1    - Input/Output.  If the D1 and D2 components of
             D2         determinant(A) = D1*2((D2 are desired, input
                        D1.GE.0.  Otherwise, input D1.LT.0.  D2 is never
                        input.

             WKAREA - Work area of length at least 2*N for IJOB=1 or 3.
                      Work area of length at least N for IJOB=2 or 4.

             IER   - Error parameter. (Output)
                     Terminal Error:
                        IER=130, Indicates that matrix A is
                           algorithmically singular.
                     Warning With Fix:
                        IER=65, Indicates that IJOB was less than 1 or
                           greater than 4.  IJOB is assumed to be 4.
```

PRECISION/HARDWARE - Single and Double/H32
- Single/H36,H48,H60

REQD IMSL ROUTINES - LUDATN, LUELMN, UERTST, UGETIO

NOTATION - Information on special notation and conventions is available in the manual introduction or through IMSL routine UHELP.

IMSL ROUTINE NAME - MDFD

PURPOSE - F probability distribution function

USAGE - CALL MDFD(F,N1,N2,P,IER)

ARGUEMENTS F - Input constant to which integration is performed. F must be greater than or equal to zero.

N1 - Input first degree of freedom. A positive integer.

N2 - Input second degree of freedom. A positive integer.

P - Output probability that a random variable following the F distribution with degrees of freedom N1 and N2 will be less than or equal to input F.

IER - Error parameter. (Output)
Terminal Error:
IER=129, Indicates either N1 or N2 is less than one or N1+N2 is greater than 20,000. P is set to positive machine infinity.
IER=130, Indicates F is less than zero. P is set to positive machine infinity.

PRECISION/HARDWARE - Single/All

REQD IMSL ROUTINES - MERRC=ERFC, UERTST, UGETIO

NOTATION - Information on special notation and conventions is available in the manual introduction or through IMSL routine UHELP.

```
IMSL ROUTINE NAME  - RLSUBM

PURPOSE            - Retrieval of a symmetric submatrix from a matrix
                     stored in symmetric storage mode by RLSTP.

USAGE              - CALL RLSUBM(A,M,IH,S,N)

ARGUEMENTS  A      - M by M symmetric matrix stored in symmetric mode.
                     (Input) A is a vector of length M*(M+1)/2.

            M      - Order of the matrix A. (Input)

            IH     - Vector of length M. (Input)
                     If IH(I)=IH(J)=1 where J and I=1,2,...,M, the
                        (I,J)-th element of A will be included in the
                        submatrix S.
                     Otherwise, the (I,J)-th element of A will not be in
                        the submatrix S on output.

            S      - Symmetric submatrix of matrix A. (Output)
                        S is a vector of length N*(N+1)/2.  A and S may
                        share the same storage if it is not necessary to
                        retain the original matrix.  See remarks for
                        RLSTP.

            N      - Order of the submatrix S. (Output)

PRECISION/HARDWARE - Single and Double/H32
                   - Single/H36,H48,H60

REQD IMSL ROUTINES - None required.

NOTATION           - Information on special notation and conventions is
                     available in the manual introduction or through
                     IMSL routine UHELP.
```

IMSL ROUTINE NAME - VCVTFS

PURPOSE - Storage mode conversion of matrices (Full to Symmetric)

USAGE - CALL VCVTFS(A,N,IA,B)

ARGUEMENTS A - Input matrix of dimension N by N. A contains a symmetric matrix stored in full mode.

N - Order of matrix A. (Input)

IA - Row dimension of matrix A exactly as specified in the dimension statement in the calling program. (Input)

B - Output vector of dimension N*(N+1)/2 containing matrix A in symmetric storage mode.

PRECISION/HARDWARE - Single and Double/H32
- Single/H36,H48,H60

REQD IMSL ROUTINES - None required.

NOTATION - Information on special notation and conventions is available in the manual introduction or through IMSL routine UHELP.

IMSL ROUTINE NAME - VCVTSF

PURPOSE - Storage mode conversion of matrices (Symmetric to Full)

USAGE - CALL VCVTSF(A,N,B,IB)

ARGUEMENTS A - Input vector of length N*(N+1)/2 containing an N by N symmetric matric stored in symmetric storage mode.

N - Order of matrix A. (Input)

B - Output matrix of dimension N by N containing matrix A in full storage mode.

IB - Row dimension of matrix B exactly as specified in the dimension statement in the calling program. (Input)

PRECISION/HARDWARE - Single and Double/H32
- Single/H36,H48,H60

REQD IMSL ROUTINES - None required.

NOTATION - Information on special notation and conventions is available in the manual introduction or through IMSL routine UHELP.

```
IMSL ROUTINE NAME  - VMULFF

PURPOSE            - Matrix multiplication (Full storage mode)

USAGE              - CALL VMULFF(A,B,L,M,N,IA,IB,C,IC,IER)

ARGUEMENTS  A      - L by M matrix stored in full storage mode. (Input)

            B      - M by N matrix stored in full storage mode. (Input)

            L      - Number of rows in A. (Input)

            M      - Number of columns in A (same as number of rows in
                        B). (Input)

            N      - Number of column in B. (Input)

            IA     - Row dimension of matrix A exactly as specified in
                        the dimension statement in the calling program.
                        (Input)

            IB     - Row dimension of matrix B exactly as specified in
                        the dimension statement in the calling program.
                        (Input)

            C      - L by N matrix containing the product C = A*B.
                        (Output)

            IC     - Row dimension of matrix C exactly as specified in
                        the dimension statement in the calling program.
                        (Input)

            IER    - Error parameter. (Output)
                     Terminal Error:
                        IER=129, Indicates A, B, or C was dimensioned
                           incorrectly.

PRECISION/HARDWARE - Single and Double/H32
                   - Single/H36,H48,H60

REQD IMSL ROUTINES - UERTST, UGETIO

NOTATION           - Information on special notation and conventions is
                     available in the manual introduction or through
                     IMSL routine UHELP.
```

```
IMSL ROUTINE NAME  - ZREAL2

PURPOSE            - Determine the real zeros of a real function - to be
                     used when initial guesses are good.

USAGE              - CALL ZREAL2(F,EPS,EPS2,ETA,NSIG,N,X,ITMAX,IER)

ARGUEMENTS  F      - A single-arguement real function subprogram
                        supplied by the user. (Input)

            EPS    - Convergence criterion. (Input)
                        A root, X(I), is accepted if ABS(X(I)).LE.EPS.

            EPS2   - Spread criteria for multiple roots. (Input)
            ETA        If the root X(I) has been computed and it is
                       found that ABS(X(I)-X(J)).LT.EPS2, where X(J) is
                       a previously computed, then the computation is
                       restarted with a guess equal to X(I)+ETA.

            NSIG   - Convergence criterion. (Input)
                        A root is accepted if two successive
                        approximations to a given root agree in the
                        first NSIG digits.

            N      - The number of roots to be found. (Input)

            X      - Vector of length N. (Input/Output)
                        On input: X contains the initial guesses for the
                           roots.
                        On output: X contains the computed roots.

            ITMAX  - Iteration indicator. (Input/Output)
                        On input: ITMAX is the maximum number of
                           iterations to be taken per root.
                        On output: ITMAX is the number of iteration used
                           in finding the last root.

            IER    - Error parameter. (Output)
                     Warning With Fix:
                        IER=33, Indicates that for one root, convergence
                           was not obtained within ITMAX iterations.
                           That root is set to 111111.
                        IER=34, Indicates that for one root, the
                           derivative of the function at that root was
                           too small.  That root is set to 222222.
                        IER=35, Indicates that the error conditions
                           described for IER=33 and IER=34 above.
                           occurred more than once.  The roots for which
                           the error occurred are set to 111111 or
                           222222, depending on the type of error.
```

PRECISION/HARDWARE - Single and Double/H32
- Single/H36,H48,H60

REQD IMSL ROUTINES - UERTST, UGETIO

NOTATION - Information on special notation and conventions is available in the manual introduction or through IMSL routine UHELP.

REMARKS 1. ZREAL2 assumes that there exist N distinct real roots for the function F and that the initial guesses supplied by the user are sufficiently close to roots to obtain convergence by Newtons method. The routine is designed so that convergence to any single root cannot be obtained from two different initial guesses. This routine is intended primarily for the refinement of N known rough approximations of the roots of F.

2. Scaling the X vector in the function F may be required if any of the roots are known to be less than one.

REFERENCES

These references are provided for those interested in further study of concepts related to those incorporated into this software. References addressed in the program comments are also provided here.

Aitkin, M.A. "Simultaneous Inference and the Choice of Variable Subsets in Multiple Regression," Technometrics, 16: 221-227 (November 1974).

Anderson, T.W. An Introduction to Multivariate Statistical Analysis (Second Edition). New York: John Wiley & Sons Inc., 1984.

Bauer, Kenneth W., Jr. Control Variate Selection for Multiresponse Simulation. Unpublished PhD dissertation. School of Industrial Engineering, Purdue University, West Lafayette IN, 1987.

Bauer, Kenneth W., Jr. and James R. Wilson. "Control-Variate Selection Criteria." Unpublished paper. 1990.

Bauer, Kenneth W., Jr., Sekhar Venkatraman, and James R. Wilson. "Estimation procedures Based on Control Variates with Known Covariance Matrix," 1987 Winter Simulation Conference Proceedings. Edited by A. Thesen, F.H. Grant, and W.D. Kelton. 334-341. New York: IEEE Press, 1987.

Cheng, R.C. and G.M. Feast. "Control Variables with Known Mean and Variance," The Journal of the Operational Research Society, 31: 51-56 (January 1980).

Furnival, G.M. and R.W. Wilson. "Regression by Leaps and Bounds," Technometrics, 16: 499-511 (November 1974).

Kennedy, W.J. and J.E. Gentle. Statistical Computing. New York: Marcel Dekker Inc., 1980.

Kleijnen, Jack P.C. Statistical Tools For Simulation Practitioners, Volume 1. New York: Marcel Dekker Inc., 1974.

Kleijnen, Jack P.C. Statistical Tools For Simulation Practitioners. New York: Marcel Dekker Inc., 1987.

Lavenberg S.S. and P.D. Welch. "A Perspective on the Use of Control Variables to Increase the Efficiency of Monte Carlo Simulations," Management Science, 27:322-335 (March 1981).

Lavenberg S.S., T.L. Moeller, and P.D. Welch. "Statistical Results on Control Variables with Application to Queueing Network Simulations," Operations Research, 30: 182-202 (1982).

Law, A.M. and W.D. Kelton. Simulation Modeling and Analysis. New York: McGraw-Hill Inc, 1982.

Mantel, N. "Why Stepdown Procedures in Variable Selection," Technometrics, 12: 621-625 (1970).

Mauro, C.A. Efficient Identification of Important Factors in Large Scale Simulation. Technical Report 123-2, Contract N00014-84-C-0573. State College PA: Desmatics Inc, October 1986 (AD-A173-497).

Nelson B.L. "Batch Size Effects on the Efficiency of Control Variates in Simulation," European Journal of Operational Research, 42: (1989).

Neter, J., W. Wasserman, and M.H. Knuter. Applied Linear Regression Models. Homewood IL: Richard D. Erwin Inc., 1983.

Pritsker, A. Alan B. Introduction to Simulation and SLAM II (Third Edition). New York: Halsted Press, 1986.

Rao, C.R. "Least Squares Theory Using an Estimated Dispersion Matrix and its Application to Management of Signals," Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Vol. I, 355-372. Berkeley CA: University of California Press, 1967.

Rubinstein, R.Y. and R. Marcus. "Efficiency of Multivariate Control Variates in Monte Carlo Simulation," Operations Research, 33, 661-677 (1985).

Ross, Sheldon M. Introduction to Probability Models (Third Edition). New York: Academic Press Inc., 1985.

Schmeiser B. "Batch Size Effects in the Analysis of Simulation Output," Operations Research, 30: 556-568 (1982).

Thompson, Mary L. "Selection of Variables in Multiple Regression: A Review and Evaluation," International Statistical Review, 5: No. 1, 37-42 (1986).

Tomick, Capt John J. A Comparison of Control Variates for Queueing Network Simulation. MS thesis, AFIT/GOR/ENS/88D-22. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1988 (AD-A202-576).

Venkatraman S. Application of the Control Variate Technique to Multiple Simulation Output Analysis. M.S. Thesis, Department of Mechanical Engineering, The University of Texas, Austin TX, 1983.

Venkatraman, Sekhar and James R. Wilson. "The Efficiency of Control Variates in Multiresponse Simulation," Operations Research Letters, 5: No. 1, 37-42 (June 1986).

Wilson, James R. "Variance Reduction Techniques for Digital Simulation," American Journal of Mathematical and Management Sciences, 4: 277-312 (March-April 1984).

Wilson, James R. and A.A.B. Pritsker. "Variance Reduction in Queueing Simulation Using Generalized Concomitant Variables," Journal of Statistical Computation and Simulation, 19: 129-153 (1984A).

Wilson, James R. and A.A.B. Pritsker. "Experimental Evaluation of Variance Reduction Techniques for Queueing Simulation Using Generalized Concomitant Variables," Management Science, 30: 1459-1472 (1984B).

APPENDIX C: Data Generation Software

SLAM Code for Simulation Model

```
5 5 25 2000. 1
100.0 0.0 1.0 2.78 2.78 25.0 25.0
0.00 1.00 0.00 0.00 0.00 0.00 0.00
0.00 0.00 1.00 0.00 0.00 0.00 0.00
0.20 0.00 0.00 0.36 0.36 0.04 0.04
0.00 0.00 1.00 0.00 0.00 0.00 0.00
0.00 0.00 1.00 0.00 0.00 0.00 0.00
0.00 0.00 1.00 0.00 0.00 0.00 0.00
0.00 0.00 1.00 0.00 0.00 0.00 0.00
GEN,BAUER,DATA GENERATION MODEL,06/05/86,1000,N,N,Y,N,N;
LIMITS,7,5,200;
STAT,1,RESPONSE TIME;
STAT,2,WAIT STAT 2;
STAT,3,WAIT STAT 3;
STAT,4,WAIT STAT 4;
STAT,5,WAIT STAT 5;
STAT,6,WAIT STAT 6;
STAT,7,WAIT STAT 7;
TIMST,XX(1),TERMINALS;
TIMST,XX(2),CPU;
TIMST,XX(3),DISK1;
TIMST,XX(4),DISK2;
TIMST,XX(5),DISK3;
TIMST,XX(6),DISK4;
TIMST,XX(7),DISK5;
INITIALIZE,0.,5000.;
MONTR,CLEAR,2000;
SEEDS,34444866917(1);
FIN;
```

FORTRAN Code for Simulation Model

```
C     PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7,TAPE1,
C    &TAPE2,TAPE3,TAPE4)
C
C ******************************************
C *  MAIN PROGRAM                          *
C ******************************************
C
      PROGRAM MAIN
      DIMENSION NSET(5000)
      COMMON QSET(5000)
      COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/UCOM1/ DEPART(10),RMEAN(10),P(10,10),SERVT(10),ECOUNT(2)
      COMMON/UCOM2/ ISUBCAP,NUSSSN,NUMCUST,TCLEAR,NSTUDY
      EQUIVALENCE (NSET(1),QSET(1))
      NNSET=5000
      NCRDR=5
      NPRNT=6
      NTAPE=7

      READ (NCRDR,*) ISUBCAP,NUSSSN,NUMCUST,TCLEAR,NSTUDY
      READ (NCRDR,*) (RMEAN(I),I=1,NUSSSN+2)

      DO 10 I=1,NUSSSN+2
         READ (NCRDR,*) (P(I,J),J=1,NUSSSN+2)
10    CONTINUE

      OPEN (UNIT=10,FILE='DGM.OP',STATUS='NEW')

      CALL SLAM

      CLOSE(10)

      STOP
      END
C
C ******************************************
C *  SUBROUTINE EVENT                      *
C ******************************************
C
      SUBROUTINE EVENT(I)
      COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/UCOM1/ DEPART(10),RMEAN(10),P(10,10),SERVT(10),ECOUNT(2)
      COMMON/UCOM2/ ISUBCAP,NUSSSN,NUMCUST,TCLEAR,NSTUDY

      ECOUNT(1)=ECOUNT(1)+1
      IF (TNOW.GT.TCLEAR) ECOUNT(2)=ECOUNT(2)+1
```

```
      GOTO (1,2),I

1     CALL ARSS
      RETURN
2     CALL ENDSS
      RETURN

      END
C
C ********************************************
C *  SUBROUTINE INTLC                        *
C ********************************************
C
      SUBROUTINE INTLC
      COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/UCOM1/ DEPART(10),RMEAN(10),P(10,10),SERVT(10),ECOUNT(2)
      COMMON/UCOM2/ ISUBCAP,NUSSSN,NUMCUST,TCLEAR,NSTUDY
      COMMON/GCOM5/ IISED(10),JJBEG,JJCLR,MMNIT,MMON,NNAME(5),NNCFI,
     &NNDAY,NNPT,NNPRJ(5),NNRNS,NNSTR,NNYR,SSEED(10),LSEED(10)
      COMMON/UCOM3/ MULTINO(7)
      INTEGER ISEED(2000)

      IF (NNRUN.EQ.1) THEN
         DO 10 I=1,2000
            ISEED(I)=(1.E+12)*DRAND(1)
10       CONTINUE
      ENDIF
      IISED(2)=ISEED(NNRUN)
      X=DRAND(-2)

      DO 15 I=1,7
         MULTINO(I)=0
15    CONTINUE

      DO 20 I=1,2
         ECOUNT(I)=0.
20    CONTINUE

      DO 25 I=1,NUSSSN+3
         DEPART(I)=0.
25    CONTINUE

      DO 30 I=1,NUSSSN+2
         SERVT(I)=0.
30    CONTINUE

      DO 35 I=1,NUMCUST
         ETIME=EXPON(RMEAN(1),2)
         ATRIB(1)=ETIME
         ATRIB(3)=I
         ATRIB(4)=1
```

```
         ATRIB(5)=2
         CALL SCHDL(1,ETIME,ATRIB)
35    CONTINUE

      DO 40 I=1,NUSSSN+2
         XX(I)=0.
40    CONTINUE

      WRITE(6,100) NNRUN,NSTUDY
100   FORMAT(1X,'SIMULATION STUDY IN PROGRESS : RUN ',I4, ' OF
     &'I4,' RUNS')
      RETURN
      END
C
C ******************************************
C *  SUBROUTINE ENDSS                       *
C ******************************************
C
      SUBROUTINE ENDSS
      COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/UCOM1/ DEPART(10),RMEAN(10),P(10,10),SERVT(10),ECOUNT(2)
      COMMON/UCOM2/ ISUBCAP,NUSSSN,NUMCUST,TCLEAR,NSTUDY
      COMMON/UCOM3/ MULTINO(7)

      CALL SCHDL(1,0.,ATRIB)
      MYQ=ATRIB(4)

      IF (NNQ(MYQ).NE.0) THEN
          CALL RMOVE(1,MYQ,ATRIB)
          WAIT=TNOW-ATRIB(2)
          CALL COLCT(WAIT,MYQ)
          RM=RMEAN(MYQ)
          SERVICE=EXPON(RM,2)
          ATRIB(4)=ATRIB(5)
          IAT=ATRIB(4)+.00001
          CALL NEXTGUY(IAT,INEXT)
C
C *********************************
C *  COLLECT STATISTICS WHILE      *
C *  PARKED AT CPU                 *
C *********************************
C
         IF (IAT.EQ.3) THEN
            MULTINO(INEXT)=MULTINO(INEXT)+1
         ENDIF

         ATRIB(5)=INEXT
         CALL SCHDL(2,SERVICE,ATRIB)
         IF (TNOW.GT.TCLEAR) THEN
            SERVT(MYQ)=SERVT(MYQ)+SERVICE
            DEPART(MYQ)=DEPART(MYQ)+1
```

```
            DEPART(NUSSSN+3)=DEPART(NUSSSN+3)+1
         ENDIF
      ELSE
         XX(MYQ)=0.
      ENDIF

      IF ((MYQ.EQ.3).AND.(NNQ(2).GT.0).AND.(ISUBCAP.NE.0).AND.
     &(INEXT.EQ.1).AND.(NNQ(MYQ).NE.0)) THEN
         CALL RMOVE(1,2,ATRIB)
         SERVICE=0.
         ATRIB(4)=ATRIB(5)
         ATRIB(5)=3
         CALL SCHDL(1,SERVICE,ATRIB)
      ENDIF

      RETURN
      END
C
C *******************************************
C *  SUBROUTINE ARSS                        *
C *******************************************
C
      SUBROUTINE ARSS
      COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/UCOM1/ DEPART(10),RMEAN(10),P(10,10),SERVT(10),ECOUNT(2)
      COMMON/UCOM2/ ISUBCAP,NUSSSN,NUMCUST,TCLEAR,NSTUDY
      COMMON/UCOM3/ MULTINO(7)

      IAT=ATRIB(5)

      IF (IAT.EQ.1) THEN
         RESP=TNOW-ATRIB(1)
         CALL COLCT(RESP,1)
         RM=RMEAN(1)
         SERVICE=EXPON(RM,2)
         ATRIB(1)=TNOW+SERVICE
         ATRIB(4)=1
         ATRIB(5)=2
         CALL SCHDL(1,SERVICE,ATRIB)
         IF (TNOW.GT.TCLEAR) SERVT(IAT)=SERVT(IAT)+SERVICE
         GO TO 101
      ENDIF

      IF (IAT.EQ.2) THEN
         IF (ISUBCAP.NE.0) THEN
            NUMSUB=0
            DO 10 I=3,NUSSSN+2
               NUMSUB=NUMSUB+NNQ(I)+XX(I)
10          CONTINUE
            IF (NUMSUB.LT.ISUBCAP) THEN
               IF (NNQ(2).EQ.0) THEN
```

```
                 WAIT=0.
                 CALL COLCT(WAIT,2)
                 SERVICE=0.
                 ATRIB(4)=2
                 ATRIB(5)=3
                 CALL SCHDL(1,SERVICE,ATRIB)
                 GO TO 101
              ELSE
                 ATRIB(2)=TNOW
                 CALL FILEM(2,ATRIB)
                 CALL RMOVE(1,2,ATRIB)
                 WAIT=TNOW-ATRIB(2)
                 CALL COLCT(WAIT,2)
                 ATRIB(4)=2
                 ATRIB(5)=3
                 SERVICE=0.
                 CALL SCHDL(1,SERVICE,ATRIB)
                 GO TO 101
              ENDIF
           ELSE
              ATRIB(2)=TNOW
              CALL FILEM(2,ATRIB)
              RETURN
           ENDIF
        ENDIF
     ENDIF

100  IF (XX(IAT).GT.0.) THEN
        ATRIB(2)=TNOW
        CALL FILEM(IAT,ATRIB)
        RETURN
     ELSE
        WAIT=0.
        CALL COLCT(WAIT,IAT)
        RM=RMEAN(IAT)
        ATRIB(4)=IAT
        CALL NEXTGUY(IAT,INEXT)
C
C *********************************
C *  COLLECT STATISTICS WHILE     *
C *  PARKED AT CPU                *
C *********************************
C
        IF (IAT.EQ.3) THEN
           MULTINO(INEXT)=MULTINO(INEXT)+1
        ENDIF
        ATRIB(5)=INEXT
        SERVICE=EXPON(RM,2)
        XX(IAT)=1
        CALL SCHDL(2,SERVICE,ATRIB)
        IF (TNOW.GT.TCLEAR) SERVT(IAT)=SERVT(IAT)+SERVICE
     ENDIF
```

```
101    IF (TNOW.GT.TCLEAR) THEN
          DEPART(IAT)=DEPART(IAT)+1
          DEPART(NUSSSN+3)=DEPART(NUSSSN+3)+1
       ENDIF

       RETURN
       END
C
C ******************************************
C *  SUBROUTINE NEXTGUY                     *
C ******************************************
C
       SUBROUTINE NEXTGUY(IAT,INEXT)
       COMMON/UCOM1/ DEPART(10),RMEAN(10),P(10,10),SERVT(10),ECOUNT(2)
       COMMON/UCOM2/ ISUBCAP,NUSSSN,NUMCUST,TCLEAR,NSTUDY

       CUM=0.
       U=UNFRM(0.,1.,2)

       DO 10 INDEX=1,NUSSSN+2
          CUM=CUM+P(IAT,INDEX)
          IF (U.LE.CUM) THEN
             INEXT=INDEX
             GOTO 15
          ELSE
             CONTINUE
          ENDIF
10     CONTINUE

15     RETURN
       END
C
C ******************************************
C *  SUBROUTINE OTPUT                       *
C ******************************************
C
       SUBROUTINE OTPUT
       COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
      1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
       COMMON/UCOM1/ DEPART(10),RMEAN(10),P(10,10),SERVT(10),ECOUNT(2)
       COMMON/UCOM2/ ISUBCAP,NUSSSN,NUMCUST,TCLEAR,NSTUDY
       COMMON/UCOM3/ MULTINO(7)

       WRITE(10,*) NNRUN
       WRITE(10,*) (ECOUNT(I),I=1,2)
       WRITE(10,*) (CCAVG(I),I=1,NUSSSN+2)
       WRITE(10,*) (TTAVG(I),I=2,NUSSSN+2)
       WRITE(10,*) (SERVT(I),I=1,NUSSSN+2)
       WRITE(10,*) (DEPART(I),I=1,NUSSSN+3)
```

```
      ISUM=0
      DO 10 I=1,7
         ISUM=ISUM+MULTINO(I)
10    CONTINUE
      WRITE(10,*) (MULTINO(I),I=1,7),ISUM

      RETURN
      END
```

FORTRAN Code for Data Post-Processing

```
      PROGRAM MKDATA
C
C *****************************************************************
C * THIS PROGRAM TAKES THE VARIABLE OUTPUTS FROM THE DATA GENERATION *
C * MODEL AND COVERTS THEM TO WORK/ROUTING VARIABLES AND RESPONSES.  *
C * THESE VARIABLES POSSESS THE DESIREABLE QUALITIES OF A MEAN OF    *
C * ZERO AND VARIANCE OF ONE.  THIS 'CONVERTED' DATA IS THEN USED AS *
C * INPUT TO THE VARIABLE SUBSET SELECTION PROGRAM.                  *
C *****************************************************************
C
      PARAMETER (NUMREPS=1000)
C
      REAL R(13),W(7),E(8)
      REAL WRKV(7)
      REAL RM(7),PI(7),YM(13)
      REAL VEC(4),RMULT(7),PI2(7)
      REAL WK(7),WK2(7)
      INTEGER MULT(8),DP
C
C *****************************************************************
C * DATA STATEMENTS                                                  *
C *                                                                  *
C *   VEC  = VECTOR OF STEADY STATE RESPONSE MEANS                   *
C *   RM   = VECTOR OF MEAN SERVICE TIMES, BY STATION                *
C *   PI   = VECTOR OF STEADY STATE TRANSITION PROBABILITIES         *
C *          (DERIVED ANALYTICALLY)                                  *
C *   PI2  = VECTOR OF ACTUAL BRANCHING PROBABILITIES FROM CPU       *
C *          (AS SUPPLIED TO THE SLAM PROGRAM)                       *
C *****************************************************************
C
      DATA VEC /30.72, 1.047, 1.458, 13.09/
      DATA RM  /100.0, 0.0, 1.0, 2.78, 2.78, 25.0, 25.0/
      DATA PI  /0.09, 0.09, 0.45, 0.16, 0.16, 0.02, 0.02/
      DATA PI2 /0.2, 0.0, 0.0, 0.36, 0.36, 0.04, 0.04/
C
C *****************************************************************
C * OPEN INPUT AND OUTPUT FILES                                      *
C *****************************************************************
C
      OPEN (UNIT=10, FILE='DGM.OP', STATUS='OLD')
      OPEN (UNIT=20, FILE='VSSP1.IN', STATUS='NEW')
      OPEN (UNIT=30, FILE='SUMMARY',  STATUS='NEW')
C
C *****************************************************************
C * READ DATA FROM FILES AND CONVERT TO WORK AND ROUTING VARIABLES   *
C *                                                                  *
C * DATA:                                                            *
C *   R1      = RUN #                                                *
C *   R2      = EVENT COUNT                                          *
```

```
C *    R3       = EVENT COUNT FROM TCLEAR                                   *
C *    R        = VECTOR OF RESPONSES                                       *
C *    W        = VECTOR OF SUMS OF SERVICE TIMES, BY STATION               *
C *    E        = VECTOR OF TOTAL DEPARTURES, BY STATION                    *
C *    MULT(J) = VECTOR OF TOTAL DEPARTURES FROM CPU TO STATION J           *
C *                                                                         *
C * OUTPUT:                                                                 *
C *    WRKV(J) = WORK VARIABLE J                                            *
C *    RMULT(J)= ROUTING VARIABLE J                                         *
C *************************************************************************
C
      DO 10 I=1,NUMREPS
C
         READ(10,*) R1
         READ(10,*) R2,R3
         READ(10,*) (R(II),II=1,7)
         READ(10,*) (R(II),II=8,13)
         READ(10,*) (W(II),II=1,7)
         READ(10,*) (E(II),II=1,8)
         READ(10,*) (MULT(II),II=1,8)
C
         DO 15 J=1,7
C
            IF (RM(J).NE.0.)THEN
               WRKV(J)=(W(J)-E(J)*RM(J))*(SQRT(E(J))
     &         /(PI(J)*E(8)*RM(J)))
            ENDIF
C
            IF ((J.EQ.2).OR.(J.EQ.3)) THEN
               RMULT(J)=0.
            ELSE
               RMULT(J)=(MULT(J)-MULT(8)*PI2(J))
     &         /SQRT(PI2(J)*(1.-PI2(J))*MULT(8))
            ENDIF
C
15       CONTINUE
C
         DO 20 J=1,13
            YM(J)=YM(J)+R(J)
20       CONTINUE
C
         DO 25 J=1,7
            WK(J)=WK(J)+WRKV(J)
            WK2(J)=WK2(J)+WRKV(J)**2
25       CONTINUE
C
C        WRITE(20,*) RMULT(1),RMULT(4),RMULT(5),RMULT(6),RMULT(7)
         WRITE(20,*) WRKV(1),WRKV(4),WRKV(5),WRKV(6),WRKV(7),
     &               R(1),R(9),R(10),R(11),R(12),R(13)
C
10    CONTINUE
C
```

```
C ********************************************************************
C * CALCULATE THE MEANS OF THE RESPONSES (YM), AND THE MEANS (WK)    *
C * AND VARIANCES (WK2) OF THE WORK VARIABLES.  THEN PRINT SUMMARY   *
C * INFORMATION TO A SUMMARY FILE.                                   *
C ********************************************************************
C
      DO 30 J=1,13
         YM(J)=YM(J)/(FLOAT(NUMREPS))
30    CONTINUE
C
      DO 35 J=1,7
         WK(J)=WK(J)/(FLOAT(NUMREPS))
         WK2(J)=WK2(J)/(FLOAT(NUMREPS))
35    CONTINUE
C
      DO 40 J=1,7
         WK2(J)=WK2(J)-WK(J)**2
40    CONTINUE
C
      WRITE(30,555)
      WRITE(30,556) NUMREPS
      WRITE(30,557)
      WRITE(30,*) VEC
      WRITE(30,558)
      WRITE(30,*) YM
      WRITE(30,559)
      WRITE(30,*) WK
      WRITE(30,560)
      WRITE(30,*) WK2
C
555   FORMAT(1X,'SUMMARY FILE OF DGM.OP DATA POST PROCESSING'/)
556   FORMAT(1X,'VSSP1.IN HAS A TOTAL OF ',I5,' REPLICATIONS'/)
557   FORMAT(1X,'BELOW ARE THE POPULATION MEANS OF THE RESPONSES'/)
558   FORMAT(1X,'BELOW ARE THE SAMPLE MEANS OF THE RESPONSES'/)
559   FORMAT(1X,'BELOW ARE THE MEANS OF THE WORK VARIABLES'/)
560   FORMAT(1X,'BELOW ARE THE VARIANCES OF THE WORK VARIABLES'/)
C
      CLOSE(10)
      CLOSE(20)
      CLOSE(30)
C
      STOP
      END
```

BIBLIOGRAPHY

Anderson, T.W. An Introduction to Multivariate Statistical Analysis (Second Edition). New York: John Wiley & Sons Inc., 1984.

Bauer, Kenneth W., Jr. Class handout distributed in OPER 766, Advanced Topics in Simulation. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, August 1989A.

Bauer, Kenneth W., Jr. 'SCOTT.II.F.' Unpublished software (FORTRAN). School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 1989B.

Bauer, Kenneth W., Jr. Control Variate Selection for Multiresponse Simulation. Unpublished PhD dissertation. School of Industrial Engineering, Purdue University, West Lafayette IN, 1987.

Bauer, Kenneth W., Jr. and James R. Wilson. Control-Variate Selection Criteria. Unpublished working paper, No. WP90-1. School of Engineering, Department of Operational Sciences, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 1990.

Bauer, Kenneth W., Jr., Sekhar Venkatraman, and James R. Wilson. 'Estimation procedures Based on Control Variates with Known Covariance Matrix,' 1987 Winter Simulation Conference Proceedings. Edited by A. Thesen, F.H. Grant, and W.D. Kelton. 334-341. New York: IEEE Press, 1987.

Furnival, G.M. and R.W. Wilson. 'Regression by Leaps and Bounds,' Technometrics, 16: 499-511 (November 1974).

Kleijnen, Jack P.C. Statistical Tools For Simulation Practitioners, Volume 1. New York: Marcel Dekker Inc., 1974.

Kleijnen, Jack P.C. Statistical Tools For Simulation Practitioners. New York: Marcel Dekker Inc., 1987.

Lavenberg S.S., T.L. Moeller, and P.D. Welch. 'Statistical Results on Control Variables with Application to Queueing Network Simulations,' Operations Research, 30: 182-202 (1982).

Law, A.M. and W.D. Kelton. Simulation Modeling and Analysis. New York: McGraw-Hill Inc, 1982.

Nelson B.L. 'Batch Size Effects on the Efficiency of Control Variates in Simulation,' European Journal of Operational Research, 42: (1989).

Neter, J., W. Wasserman, and M.H. Knuter. Applied Linear Regression Models. Homewood IL: Richard D. Erwin Inc., 1983.

Pritsker, A. Alan B. Introduction to Simulation and SLAM II (Third Edition). New York: Halsted Press, 1986.

Rao, C.R. "Least Squares Theory Using an Estimated Dispersion Matrix and its Application to Management of Signals," Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Vol. I, 355-372. Berkeley CA: University of California Press, 1967.

Rubinstein, R.Y. and R. Marcus. "Efficiency of Multivariate Control Variates in Monte Carlo Simulation," Operations Research, 33, 661-677 (1985).

Ross, Sheldon M. Introduction to Probability Models (Third Edition). New York: Academic Press Inc., 1985.

Schmeiser B. "Batch Size Effects in the Analysis of Simulation Output," Operations Research, 30: 556-568 (1982).

Tomick, Capt John J. A Comparison of Control Variates for Queueing Network Simulation. MS thesis, AFIT/GOR/ENS/88D-22. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1988 (AD-A202-576).

Wilson, James R. "Variance Reduction Techniques for Digital Simulation," American Journal of Mathematical and Management Sciences, 4: 277-312 (March-April 1984).

VITA

Captain James A. Gigliotti [redacted], [redacted]. He graduated from high school in Erie, Pennsylvania, in 1978 and attended Carnegie-Mellon University, from which he received the degree of Bachelor of Science in Mechanical Engineering in May 1982. After graduation, he received a commission in the USAF through the Officer Training School. He has served as a Space Shuttle Support Equipment Manager, Space Shuttle Launch Facility Test Preparation Manager, and Space Shuttle Program Requirements Document Manager as part of the Shuttle Activation Task Force (SATAF), and later, the 6595th Shuttle Test Group (STG) at Vandenberg AFB, CA. He next served as Chief, ASAT Product Assurance Branch at Los Angeles AFB, CA, until entering the School of Engineering, Air Force Institute of Technology, in August 1988.

[redacted]

[redacted]

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GOR/ENS/90M-7	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION AFIT 6b OFFICE SYMBOL (If applicable) ENS	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Wright-Patterson AFB, OH 45433	7b ADDRESS (City, State, and ZIP Code)
8a NAME OF FUNDING/SPONSORING ORGANIZATION AFIT 8b OFFICE SYMBOL (If applicable) ENS	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code) Wright-Patterson AFB, OH 45433	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification)
Comparison of Selection Procedures and Validation of Criterion Used in Selection of Significant Control Variates of a Simulation Model (U)

12. PERSONAL AUTHOR(S)
James A. Gigliotti, Captain, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Masters Thesis	FROM ______ TO ______	1990 March	163

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
12	03	
12	05	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Control Variables   Selection Criterion   Simulation
Statistical Analysis   Multivariate Analysis
Computer Application   Correlation Technique

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The purpose of this thesis was to revise and extend the capabilities of existing software for selecting significant control variates of a simulation model using a newly developed selection criterion, test the software on prepared data, and validate the selection criterion.

The revised software is called the Variable Subset Selection Program (VSSP). The VSSP was tested using two sets of data. The first data set was derived using a known stochastic structure and the second data set was created using an untested simulation model. These results were also compared to variable subsets found using commercial software employing more common selection criteria.

The results obtained from this thesis served to demonstrate the usefullness of the VSSP and the validity of the new selection criterion.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL James A. Gigliotti, Captain, USAF	22b. TELEPHONE (Include Area Code) (513) 255-5758 22c. OFFICE SYMBOL AFIT/ENS